

# (12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织  
国际局

(43) 国际公布日  
2017年11月30日 (30.11.2017)



(10) 国际公布号  
WO 2017/202373 A1

- (51) 国际专利分类号:  
*H04N 21/654* (2011.01) *H04N 21/658* (2011.01)
- (21) 国际申请号: PCT/CN2017/086012
- (22) 国际申请日: 2017年5月25日 (25.05.2017)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (30) 优先权:  
201610355474.7 2016年5月25日 (25.05.2016) CN
- (71) 申请人: 中兴通讯股份有限公司 (ZTE CORPORATION) [CN/CN]; 中国广东省深圳市南山区高新技术产业园科技南路中兴通讯大厦, Guangdong 518057 (CN)。
- (72) 发明人: 许智敏 (XU, Zhimin); 中国广东省深圳市南山区高新技术产业园科技南路中兴通讯大厦, Guangdong 518057 (CN)。 刘利 (LIU, Li); 中国广东省深圳市南山区高新技术产业园科技南路中兴通讯大厦, Guangdong 518057 (CN)。 张行功 (ZHANG, Xingong); 中国广东省深圳市

南山区高新技术产业园科技南路中兴通讯大厦, Guangdong 518057 (CN)。 董振江 (DONG, Zhenjiang); 中国广东省深圳市南山区高新技术产业园科技南路中兴通讯大厦, Guangdong 518057 (CN)。 黄成 (HUANG, Cheng); 中国广东省深圳市南山区高新技术产业园科技南路中兴通讯大厦, Guangdong 518057 (CN)。

(74) 代理人: 北京康信知识产权代理有限责任公司 (KANGXIN PARTNERS, P. C.); 中国北京市海淀区知春路甲48号盈都大厦A座16层, Beijing 100098 (CN)。

(81) 指定国 (除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM,

(54) Title: STREAMING MEDIA QUICK START METHOD, DEVICE AND SYSTEM

(54) 发明名称: 流媒体快速启动方法、装置和系统

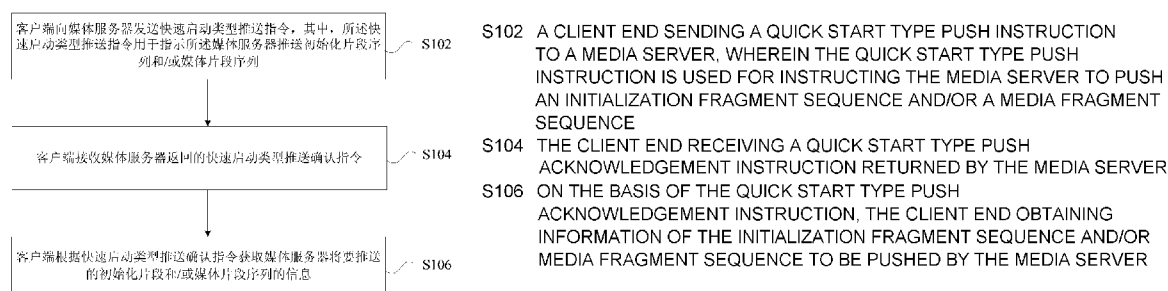


图 1

(57) Abstract: Provided in the present disclosure are a streaming media quick start method, device and system. The method comprises: a client end sending a quick start type push instruction to a media server, wherein the quick start type push instruction is used for instructing the media server to push an initialization fragment sequence and/or a media fragment sequence; the client end receiving a quick start type push acknowledgement instruction returned by the media server; on the basis of the quick start type push acknowledgement instruction, the client end obtaining information of the initialization fragment sequence and/or media fragment sequence to be pushed by the media server. The present disclosure solves the problem wherein a media client end and a media server have not negotiated adequate synchronization, and in turn achieves the effect of the media client end being capable of determining whether an expected resource is in a push list, such that a next decision may be made as soon as possible and a quick start may be better achieved.

ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US,  
UZ, VC, VN, ZA, ZM, ZW。

- (84) 指定国(除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

- 包括国际检索报告(条约第21条(3))。

---

(57) 摘要: 本公开提供了一种流媒体快速启动方法、装置和系统。其中, 该方法包括: 客户端向媒体服务器发送快速启动类型推送指令, 其中, 快速启动类型推送指令用于指示媒体服务器推送初始化片段序列和/或媒体片段序列; 客户端接收媒体服务器返回的快速启动类型推送确认指令; 客户端根据快速启动类型推送确认指令获取媒体服务器将要推送的初始化片段和/或媒体片段序列的信息。通过本公开, 解决了媒体客户端与媒体服务器端之间没有达到很好的协商同步的问题, 进而达到了媒体客户端可以判断自己期望的资源是否在推送列表, 以便尽早做出下一步决策, 实现更好的快速启动的效果。

## 流媒体快速启动方法、装置和系统

### 技术领域

本公开涉及流媒体领域，具体而言，涉及一种流媒体快速启动方法、装置和系统。

### 背景技术

随着网络的发展和 HTTP 流媒体技术的普及，越来越多的媒体内容提供者采用 HTTP 动态自适应流媒体系统为用户提供高质量的流媒体服务，媒体内容可为音频、视频、文本及其组合等。HTTP 动态自适应流媒体(DASH)系统不仅可以支持多种编码、解码技术及标准，还可以支持不同的媒体格式和媒体播放器，同时还可以提供多种视角、码率、帧率、空间分辨率及其组合等的流媒体资源。

在 HTTP 动态自适应流媒体(DASH)系统中，媒体内容可包括媒体呈现描述文件(MPD-Media Presentation Description)和媒体片段(Segments)资源。媒体呈现描述文件是描述媒体内容的可扩展标记语言(XML-Extensible Markup Language)文档，其中包含媒体片段资源的描述信息，比如媒体片段的呈现时间、码率、帧率、空间分辨率、统一资源定位符、以及所在的时段、适配集、表述等信息。媒体内容可以分为多个层级，包括时段(Period)、适配集(AdaptationSet)、表述(Representation)以及片段(Segment)。

在 HTTP 动态自适应流媒体系统中，客户端通过向服务器端请求媒体呈现描述文件来开启一个流媒体会话(Streaming Session)。当收到媒体呈现描述文件之后，客户端会分析媒体呈现描述文件，选择一个适合客户端环境的适配集，然后在该适配集中选择一个适合客户端带宽、解码和展示的表述。最后，客户端会在选定的表述中向服务器请求适合客户端环境的初始化片段，并通过 MPD 文件的对应描述信息来依次请求初始化片段对应的片段序列，然后展示媒体资源。需要注意的是，客户端会在真正开始展示前请求并缓存一定数量的媒体数据。而每个资源文件的请求/响应(Requests/Responses)和缓存的填充等都会为流媒体会话引入启动延迟(Startup Delay)。一方面，这给用户带来了不好的观看体验，比如在一个直播中，存在很大的直播延迟，影响观看体验。另一方面，给媒体内容提供者带来了流失客户的风险，比如客户在还没有观看媒体内容提供者提供的资源时，就离开了媒体内容提供者提供的服务。所述启动延迟(Startup Delay)指的是从发送媒体呈现描述文件请求到展示第一个资源画面，比如第一个视频帧，之间的时间消耗值，它与网络回环(Round-trip)、缓存大小和媒体资源片段的大小等因素有关。

全双工 HTTP 动态自适应流媒体(FDH-DASH)通过利用 FDH 服务器推送(Server Push)技术，可以在客户端请求媒体呈现描述文件的同时，用来推送初始化片段和/或起始媒体片段。相比于传统的依赖于解析媒体呈现描述文件的方式，采用 FDH 服务器推送策略将允许客户端通过减少回环(Round-trip)时间，来尽早的展示媒体资源。所述片段请求/响应消息可通过 WebSocket 或 HTTP/2 等全双工 HTTP 协议发送。基于这种思想很多公司提出了自己的快速启

动(Fast Start)方案。

一种快速启动(Fast Start)策略是通过客户端向服务器发送带有快速启动推送指令(push fast-start directive)的媒体呈现描述文件的请求(request)消息,告诉服务器客户端希望在获取媒体呈现描述文件的同时,获取一些或者全部可用的初始化片段(Initialization Segments)。所述 fast-start directive 不带参数,仅由快速启动信息(fast-start)组成。服务器收到上述消息后,首先发送包含不带参数的快速启动推送指令的对应媒体呈现描述文件请求(request)的响应消息(response),以此来告诉客户端服务器接收快速启动的请求,并且会推送一些或者全部可用的初始化片段给客户端。然后,服务器给客户端发送一些或者全部可用的初始化片段。

另一种快速启动(Fast Start)策略通过客户端向服务器发送带有快速启动推送指令(push fast-start directive)的媒体呈现描述文件的请求消息,告诉服务器客户端希望在获取媒体呈现描述文件的同时,获取一些或者全部可用的初始化片段(Initialization Segments),或者一些或者全部可用的初始化片段和指定类型(Video 或 Audio)、指定数量 L (亦可不指定)的媒体片段,或者一些或者全部可用的初始化片段、指定数量 M (亦可不指定)的 Video 媒体片段和指定数量 N(亦可不指定)的 Audio 媒体片段等,在指定的同时还可指定语言(Language),比如'lang=ja'。所述 push-fast-start directive 由快速启动信息(fast-start)和推送参数组成。所述推送参数有多种,可以为 video = M、audio = N、init-only 或者他们相互组合(video、audio 数目亦可不指定)。服务器收到上述消息后,首先发送包含快速启动推送指令的对应媒体呈现描述文件请求(request)的响应消息(response),以此来告诉客户端服务器接收快速启动的请求,并且会推送一些或者全部可用的初始化片段和快速启动推送指令指定数量中,服务器可以推送的媒体片段给客户端。然后,服务器给客户端发送初始化片段和快速启动推送指令指定数量中服务器可以推送的片段序列。

在 DASH P6 标准 (Information Technology – Dynamic adaptive streaming over HTTP(DASH) – Part 6: DASH with Server Push and WebSockets) CD 中,采用的快速启动(Fast Start)策略是通过客户端向服务器发送带有快速启动推送指令(push-fast-start directive)的媒体呈现描述文件的请求(request)消息,告诉服务器客户端希望在获取媒体呈现描述文件的同时,获取一些或者全部可用的初始化片段(Initialization Segments)。所述 push-fast-start directive 不带参数,仅由快速启动信息(push-fast-start)组成。服务器收到上述消息后,首先发送包含不带参数的快速启动推送指令的对应媒体呈现描述文件请求(request)的响应消息(response),以此来告诉客户端服务器接收快速启动的请求,并且会推送一些或者全部可用的初始化片段给客户端。然后,服务器给客户端发送一些或者全部可用的初始化片段。该 CD 中的定义类似于以上所述的第一种方案,区别仅在指令的命名。

上述的第一种、第二种与第三种快速启动(Fast Start)策略,客户端不知道服务器会确切的推送多少个初始化片段,也不知道何时会结束推送,这造成了客户端和服务器之间,时间资源的浪费,与快速启动策略的设计理念相悖。而上述的第二种快速启动策略,定义了很多消息参数,虽然比较详细,但是有些累赘,比如如果 MPD 描述文件中根本不存在 audio 资源,客户端盲目指定会造成请求头部资源的开销,这与 DASH 的设计理念相悖,而且,该策略为

客户端驱动的快速启动策略，服务器根据客户端的偏好，决定所推送的初始化片段，这显然与客户端需要在 MPD 文件请求时发送推送偏好信息相悖，首先，快速启动时客户端不知道 MPD 文件具体描述信息，其次即使根据经验，也还是会有命中失败(Miss)的情况发生。如果快速启动策略完全由服务器驱动，正如上述第一种和第三种快速启动策略，不仅客户端不知道服务器推送哪些内容，更不知道何时推送结束，媒体客户端与媒体服务器端之间没有达到很好的协商同步。

## 发明内容

本公开提供了一种流媒体快速启动方法、装置及系统，以至少解决相关技术中媒体客户端与媒体服务器端之间没有达到很好的协商同步的问题。

根据本公开的一个方面，提供了一种流媒体快速启动方法，包括：客户端向媒体服务器发送快速启动类型推送指令，其中，快速启动类型推送指令用于指示媒体服务器推送初始化片段序列和/或媒体片段序列；客户端接收媒体服务器返回的快速启动类型推送确认指令；客户端根据快速启动类型推送确认指令获取媒体服务器将要推送的初始化片段和/或媒体片段序列的信息。

根据本公开的另一个方面，还提供了一种流媒体快速启动方法，包括：媒体服务器接收客户端发送的快速启动类型推送指令，其中，快速启动类型推送指令用于指示媒体服务器推送初始化片段序列和/或媒体片段序列；媒体服务器根据将要推送的初始化片段和/或媒体片段序列，生成快速启动类型推送确认指令；媒体服务器向客户端发送快速启动类型推送确认指令。

根据本公开的又一方面，提供了一种流媒体快速启动装置，包括：访问引擎模块，设置为客户端向媒体服务器发送快速启动类型推送指令，其中，快速启动类型推送指令用于指示媒体服务器推送初始化片段序列和/或媒体片段序列；第一接收模块，设置为客户端接收媒体服务器返回的快速启动类型推送确认指令；控制引擎模块，设置为客户端根据快速启动类型推送确认指令获取媒体服务器将要推送的初始化片段和/或媒体片段序列的信息。

根据本公开的再一方面，还提供了一种流媒体快速启动装置，包括：服务模块，设置为媒体服务器接收客户端发送的用于获取媒体呈现描述文件 MPD 的请求消息，其中，请求消息至少包括：快速启动类型推送指令和 MPD 请求指令；生成引擎模块，媒体服务器根据请求消息生成响应信息，其中，响应消息包括：确定客户端需要快速启动的推送确认指令、响应 MPD 请求指令得到的 MPD 数据；解析引擎模块，设置为在媒体服务器生成预测的片段消息之后，媒体服务器将预测的片段消息和响应信息推送至客户端，使得客户端根据预测的片段消息和 MPD 数据，获取客户端需要的媒体内容，其中，预测的片段消息为媒体服务器主动推送的与推送确认指令对应的片段消息。

根据本公开的第五个方面，还提供了一种流媒体快速启动系统，包括：客户端，生成快速启动推动指令，并向媒体服务器发送用于获取媒体呈现描述文件 MPD 的请求消息至媒体服

务器，在获取到预测的片段消息之后，根据预测的片段消息和 MPD 数据，获取需要的媒体内容，其中，请求信息至少包括：快速启动类型推送指令和 MPD 请求指令，预测的片段消息为媒体服务器主动推送的与推送确认指令对应的片段消息。媒体服务器，与客户端通信，设置为接收客户端发送的用于获取媒体呈现描述文件 MPD 的请求消息，并根据请求消息生成响应信息，在生成预测的片段消息之后，将预测的片段消息和响应信息推送至客户端。

根据本公开的第六个方面，还提供了一种存储介质，所述存储介质包括存储的程序，其中，所述程序运行时执行上述任一项所述的方法。

通过本公开，采用客户端通过生成所述 fast-start 类型推送指令，即可告诉媒体服务器，客户端期待媒体服务器推送片段序列，以快速启动；通过使用从所述媒体服务器接收服务器推送的 K 个片段的片段消息，减少了客户端盲目等待的时间，降低了启动延迟，解决了媒体客户端与媒体服务器端之间没有达到很好的协商同步的问题，进而达到了媒体客户端可以判断自己期望的资源是否在推送列表，以便尽早做出下一步决策，实现更好的快速启动的效果。

## 附图说明

此处所说明的附图用来提供对本公开的进一步理解，构成本申请的一部分，本公开的示意性实施例及其说明用于解释本公开，并不构成对本公开的不当限定。在附图中：

图 1 是根据本公开实施例的一种流媒体快速启动方法的流程图；

图 2 是 FDH-DASH 系统架构 100 的一个实施例的示意图；

图 3 示出了一个 FDH 媒体客户端的实施例的示意图；

图 4 示出了一个 FDH 媒体服务器的实施例的示意图；

图 5A-D 示出了推送参数为 NUMBER 媒体呈现 MPD 与其请求响应，以及媒体客户端与媒体服务器是实施例交互流程示意图；

图 6 是由用户端设备实现的推送参数为 NUMBER 实施例的流程图；

图 7A-D 示出了推送参数为 NUMBER 媒体呈现 MPD 与其请求响应，以及媒体客户端与媒体服务器是实施例交互流程示意图；

图 8 是根据本公开实施例的一种流媒体快速启动方法的流程图；

图 9A-D 示出了推送参数为 URL\_LIST 方法媒体呈现 MPD 与其请求响应，以及媒体客户端与媒体服务器是实施例交互流程示意图；

图 10A-D 示出了推送参数为 URL\_TEMPLATE 方法媒体呈现 MPD 与其请求响应，以及

媒体客户端与媒体服务器是实施例交互流程示意图；

图 11 是根据本公开实施例的一种流媒体快速启动方法的流程图；

图 12 是根据本公开实施例的一种流媒体快速启动方法的流程图；

图 13 是计算机系统或网络设备 1200 的实施例的示意图；

图 14 是根据本公开实施例的一种流媒体快速启动方法的流程图；

图 15 是根据本申请实施例的一种可选的流媒体快速启动方法；

图 16 是根据本公开实施例的一种流媒体快速启动方法的流程图；

图 17 是根据本公开实施例的一种流媒体快速启动装置的示意图；

图 18 是根据本公开实施例的另一种流媒体快速启动装置的示意图；以及

图 19 是根据本公开实施例的另一种流媒体快速启动系统的示意图。

## 具体实施方式

下文中将参考附图并结合实施例来详细说明本公开。需要说明的是，在不冲突的情况下，本申请中的实施例及实施例中的特征可以相互组合。

需要说明的是，本公开的说明书和权利要求书及上述附图中的术语“第一”、“第二”等是用于区别类似的对象，而不必用于描述特定的顺序或先后次序。

还需要说明的是，该公开可用于 DASH 网络，特别是 FDH-DASH 网络，且不限于上述领域。

需要说明的是，下文中“快速启动推送指令”和“快速启动类型推送指令”指的均是“类型”字段为 fast\_start 的推送指令。

下面对于实施例中出现的专业术语进行解释：

FDH(Full Duplex HTTP-compatible protocols)是指与 HTTP 1.1 标准兼容，且支持全双工通信的协议，比如 WebSocket 和 HTTP/2 等协议的全双工超文本协议。

DASH(Dynamic Adaptive Streaming over HTTP)是基于 HTTP 的媒体内容流传输的通信标准。FDH-DASH (DASH over Full Duplex HTTP compatible protocols)即基于全双工超文本协议的 DASH 技术标准。

在本实施例中提供了一种流媒体快速启动方法，图 1 是根据本公开实施例的一种流媒体快速启动的流程图，如图 1 所示，该流程包括如下步骤：

步骤 S102，客户端向媒体服务器发送快速启动类型推送指令，其中，所述快速启动类型

推送指令用于指示所述媒体服务器推送初始化片段序列和/或媒体片段序列。

步骤 S104, 所述客户端接收所述媒体服务器返回的快速启动类型推送确认指令。

步骤 S106, 所述客户端根据所述快速启动类型推送确认指令获取所述媒体服务器将要推送的初始化片段和/或媒体片段序列的信息。

在一种可选的实施例中, 客户端生成快速启动类型推送指令 `push-fast-start directive`, 向媒体服务器发送所述快速启动类型推送指令, 其中, 所述快速启动类型推送指令用于通知所述媒体服务器推送初始化片段序列和/或媒体片段序列, 所述客户端获取快速启动类型推送确认指令 `push-fast-start acknowledge`, 所述客户端根据所述快速启动类型推送确认指令获取所述服务器将要推送的初始化片段和媒体片段序列的信息。

在一种可选的实施例中, 上述流媒体快速启动方法可以包括如下步骤:

客户端生成快速启动推动指令。

客户端发送用于获取媒体呈现描述文件 `MPD` 的请求消息至媒体服务器, 其中, 请求信息至少包括: 快速启动推送指令和 `MPD` 请求指令。

在上述步骤中, 请求消息可以为 `MPD` 片段请求消息, 客户端生成快速启动推送指令 `push-fast-start directive`, 向媒体服务器发送媒体呈现描述文件 (`MPD`) 的请求消息, 其中, 所述媒体呈现描述文件的请求由上述快速启动推送指令 `push-fast-start directive` 和请求内容组成。

客户端接收媒体服务器根据请求消息返回的响应消息, 其中, 响应消息包括: 确定客户端需要快速启动的推送确认指令、响应 `MPD` 请求指令得到的 `MPD` 数据。

步骤 S108, 在客户端获取到预测的片段消息之后, 根据预测的片段消息和 `MPD` 数据, 获取客户端需要的媒体内容, 其中, 预测的片段消息为媒体服务器主动推送的与推送确认指令对应的片段消息。

在上述步骤中, 客户端从所述媒体服务器接收响应所述片段消息的响应, 以及媒体服务器主动推送的初始化片段和/或片段序列, 其中, 所述快速启动推送指令 `push-fast-start directive` 由快速启动消息和一系列参数组成。流媒体客户端通过解析接收到的响应信息快速启动推送指令 `push-fast-start directive` 确认指令中的参数, 并对参数列表进行解析, 根据下一步所需内容是否在媒体服务器推送片段列表, 来快速决定下一步媒体客户端要做的具体操作, 如果媒体服务器推送命中失败 (`Miss`), 亦即所需内容不再推送列表中, 可以立即发送对应所需消息的请求, 也可以取消 (`Cancel`) 媒体服务器的推送, 否则直接获取所需内容。

通过本公开, 采用客户端通过生成所述 `push-fast-start directive`, 即可告诉媒体服务器, 客户端期待媒体服务器推送片段序列, 以快速启动; 通过使用从所述媒体服务器接收服务器推送的 `K` 个片段的片段消息, 减少了客户端盲目等待的时间, 降低了启动延迟, 解决了媒体客户端与媒体服务器端之间没有达到很好的协商同步的问题, 进而达到了媒体客户端可以判断自己期望的资源是否在推送列表, 以便尽早做出下一步决策, 实现更好的快速启动的效果。

通过以上步骤，媒体服务器只需根据片段请求消息中的快速启动(**fast-start**)信息，通过媒体服务器向媒体客户端发送推送确认指令，并主动推送由服务器驱动决策的一系列预测片段给媒体客户端，媒体客户端在收到媒体呈现描述文件的响应消息之后，会解析媒体响应消息的头部信息，通过对该消息中的推送确认指令的解析，获取媒体服务器推送数目 **K** 和要推送的 **K** 个片段的内容信息，比如 **URL** 等信息。通过这些信息，媒体客户端可以判断自己期望的资源是否在推送列表，以便尽早做出下一步决策，实现更好的快速启动，支持快速启动(**Fast Start**)低延迟的设计理念。

在本公开介绍的实施例中，媒体客户端根据是否需要快速启动和 **MPD** 信息生成 **push-fast-start directive**，其中的 **push-fast-start directive** 可以不包含任何参数，也可以包含一些表示客户端偏好的参数，比如加入客户端偏好的码率，偏好的媒体片段类型，偏好的媒体片段个数等或这些的组合等，媒体服务器根据收到的 **MPD** 文件请求中是否包含 **push-fast-start directive** 确定客户端需要快速启动，然后媒体服务器通过本地决策（或者通过结合媒体客户端推送的 **push-fast-start directive** 参数，亦即客户端偏好来决策）推送的片段数目，并决策要推送的具体片段。并通过 **MPD** 响应消息 **push-fast-start acknowledge** 来通知媒体客户端，其中 **push-fast-start acknowledge** 由快速启动信息(**fast-start**)和参数组成，同时在发送 **MPD** 文件的同时，推送决策生成的片段序列。媒体客户端在收到 **push-fast-start acknowledge** 后，解析获取所请求的 **MPD** 文件响应头部信息中是否含有 **push-fast-start acknowledge**，并获取其中的参数，通过查询媒体服务器发送过来的片段序列是否符合媒体客户端的需求，来快速进行下一步决策。命中失败(**Miss**)时，可以立即发送对应所需消息的请求，也可以取消(**Cancel**)媒体服务器的推送，立即请求媒体客户端需要的文件，命中时，直接获取所需的文件。

需要明白的一点是，所有快速启动策略均在媒体客户端请求 **MPD** 文件时进行，命中失败(**Miss**)的情形均可能发生。本公开所提快速启动策略，相比于已经提出的与 **FDH** 标准 **CD** 中的快速启动策略，实现了媒体客户端对于推送文件的透明性，亦即媒体客户端可以通过媒体服务器发送的 **push-fast-start acknowledge** 中的参数与媒体服务器发送的片段序列明确媒体服务器是否命中，以及是否该继续等待媒体服务器推送内容的到达，从一定程度上，减少了媒体客户端盲目等待时间，有效的支持快速启动策略设计理念。

图 2 是 **FDH-DASH** 系统架构 100 的一个实施例的示意图。**FDH-DASH** 系统 100 包括一个或多个媒体客户端 110，所述媒体客户端 110 包括用户设备或装置，如机顶盒 (**STB**)、台式电脑、笔记本电脑、平板电脑、智能手机、任意其它类型的移动设备、个人设备或这些设备的组合等，所述媒体客户端 110 支持全双工超文本传输协议(**FDH**)。**FDH-DASH** 架构 100 包括一个或多个媒体服务器 120，所述媒体服务器可以是任何合适类型的网络服务器，如内容分发网络(**CDN**)的一个节点、代理服务器、**Web** 服务器、或这些服务器的组合等，所述媒体服务器 120 支持全双工超文本传输协议(**FDH**)。所述媒体客户端 110 可以通过任意接入网 130 与所述媒体服务器 120 建立 **FDH** 连接，所述接入网 130 包括互联网、无线局域网(**WLAN**)、**WiFi** 网络或热点、蜂窝网、线缆架构、光纤骨干网、接入网或这些网络的组合等。所述媒体客户端 110 与所述媒体服务器 120 通过 **FDH** 连接，它们之间可以交换请求/响应消息，发送/接收媒体内容。

图 3 示出了一个 FDH 媒体客户端的实施例的示意图。如图 3 所示，所述媒体客户端 110 包括 FDH-DASH 控制引擎 111、生成引擎 112、FDH 访问引擎 113 和媒体引擎 114。

FDH-DASH 控制引擎 111，可以解析 MPD，获得关于所述媒体内容的信息，如内容时序、内容或媒体可用性、媒体类型、分辨率、最小及最大带宽、媒体内容在网络中的位置及其它内容特征，如媒体呈现时间线、媒体片段列表等；可以根据用户设备的处理能力、显示能力和客户端缓存变化趋势、可用带宽的变化趋势，决定所需下载/推送的片段序列，例如，在可用带宽大时可选择高质量的片段序列；可以根据服务器端发送的 `push-fast-start acknowledge` 中的参数和服务器推送的片段信息决定是否命中失败(Miss)，如果命中失败，可以直接忽略服务器推送的片段，也可以直接取消(Cancel)服务器推送，启动新的片段请求，否则等待获取所需内容。

生成引擎 112，根据客户端是否需要快速启动生成 `push-fast-start directive`，其中，所述 `push-fast-start directive` 可以不包含任何参数，也可以包含一些表示客户端偏好的参数，比如加入客户端偏好的码率，偏好的媒体片段类型，偏好的媒体片段个数等或这些的组合等；生成 `push-template` 类型推送指令等，所述 `push-template` 类型推送指令用于正常片段请求，具体描述可见 FDH 标准定义。

FDH 访问引擎 113，发起并建立 FDH 连接通道；根据客户端是否需要快速启动生成的 `push-fast-start directive` 生成并发送 MPD 请求消息，接收并解析 MPD 响应消息，获取 MPD 数据；根据 MPD 请求的响应消息获取服务器端发送的 `push-fast-start acknowledge` 以及该指令中的参数；接收被推送的片段序列。

媒体引擎 114，负载解码媒体片段，呈现流媒体。

图 4 示出了一个 FDH 媒体服务器的实施例的示意图。如图 4 所示，所述媒体服务器 120 包括 FDH 服务模块 121、解析引擎 122、媒体内容 130 和媒体准备单元 140。所述媒体内容 130 可以由媒体准备单元 140 生成或准备，所述媒体准备单元 140 可以位于所述媒体服务器上或在其他地方(如，位于内容提供者中)。所述媒体内容 130 可以是任何可行的形式，如视频、音频、文本及其组合等。所述媒体内容 130 可以包括媒体呈现描述(MPD)131 和多个片段资源，所述 MPD 131 为描述媒体内容的 XML 文档，包含媒体片段资源的描述信息，如媒体片段的起始呈现时间、码率、帧率、空间分辨率、统一资源定位符、以及所在的时段、适配集、表述等等信息。所述 MPD 还可以包括片段资源的 URL 信息和呈现时间信息(如，起始呈现时间、呈现时长等等)，其中，所述 URL 信息可通过片段模板(SegmentTemplate)或片段列表(SegmentList)提供，所述片段资源的呈现时间信息可通过片段时间线(SegmentTimeline)或多片段基信息(MultipleSegmentBaseInformation)等给出。所述片段的起始呈现时间是该媒体片段的帧数据对应的呈现时间，相对于该媒体片段所在时段(Period)的时段起始时间(PeriodStart time)。所述 MPD 131 可以与所述媒体内容 130 的片段资源位于相同的网络位置，也可以位于其他位置。所述媒体内容 130 可以包括一个或多个时段，且每个时段可以包括一个或多个适配集，且每个适配集可以包括一个或多个表述。例如，在第一时段 132 内，适配集 133 可以包含多个表述，表述 134 可以包含多个时段。

FDH 服务模块 121, 接收 FDH 连接请求, 建立 FDH 连接通道; 接收并解析 MPD 请求消息, 获取媒体客户端是否需要快速启动的指令; 生成并发送 MPD 响应消息, 所述 MPD 响应消息包含 MPD 数据, push-fast-start acknowledge, 以及 push-fast-start acknowledge 中的参数; 接收并解析片段请求消息; 生成并发送片段响应消息; 推送媒体服务器主动预测的片段序列。

解析引擎 122, 负责解析 MPD 请求消息, 获取媒体客户端是否需要快速启动的指令, 当媒体客户端包含 push-fast-start directive 时, 计算媒体服务端要推送的片段多少, 获取 push-fast-start acknowledge 中所指定数目的片段序列; 解析客户端的推送指令, 获取客户端需要的片段序列。

可选地, 根据本申请上述实施例, 所述 MPD 请求消息还包括: 至少一个请求参数的情况下, 所述媒体服务器将要推送的初始化片段和媒体片段序列的信息包括所述 MPD 请求消息的片段的属性; 所述客户端通过如下任意一种或多种方式从所述媒体服务器获取到所述 MPD 响应消息: HTTP/2 的 MPD 响应报文方式和 WebSocket 中 DASH 子协议的 new\_mpd 消息, 其中, HTTP/2 的 MPD 响应报文通过 Push-Policy 首部携带快速启动类型推送确认指令; WebSockets DASH 子协议的 new\_mpd 消息通过 JSON 键值对形式携带快速启动类型推送确认指令。

可选地, 在本申请上述实施例中, 所述客户端接收所述媒体服务器返回的 MPD 响应消息, 以及所述快速启动类型推送确认指令, 进一步包括:

通过 HTTP/2 报文方式接收所述 MPD 响应消息, 其中, 所述 HTTP/2 报文的 Accept-Push-Policy 首部携带快速启动类型推送确认指令; 或者,

通过 WebSocket 中 DASH 子协议的 new\_mpd 消息接收所述 MPD 响应消息, 其中, 所述 new\_mpd 消息的 JSON 键值对携带快速启动类型推送确认指令。

可选地, 在本申请上述实施例中, 所述客户端向所述媒体服务器发送 MPD 请求消息, 以及所述快速启动类型推送指令, 进一步包括:

通过 HTTP/2 报文方式发送所述 MPD 请求消息, 其中, 所述 HTTP/2 报文的 Accept-Push-Policy 首部携带快速启动类型推送指令; 或者,

通过 WebSocket 中 DASH 子协议的 get\_mpd 消息发送所述 MPD 请求消息, 其中, 所述 get\_mpd 消息的 JSON 键值对携带快速启动类型推送确认指令。

在一种可选的实施例中, 所述客户端通过如下任意一种或多种方式发送所述 MPD 请求消息: HTTP/2 请求报文方式和 WebSocket 中 DASH 子协议的 get\_mpd 消息, 其中, HTTP/2 请求报文通过 Accept-Push-Policy 首部携带快速启动类型推送指令; WebSockets 中 DASH 子协议的 get\_mpd 消息通过 JSON 键值对形式携带快速启动类型推送指令。

在另一种可选的实施例中, 在所述 MPD 请求消息还包括: 至少一个请求参数的情况下, 所述服务器将要推送的初始化片段和媒体片段序列的信息包括所述 MPD 请求消息的片段的属

性；所述客户端通过如下任意一种或多种方式从所述媒体服务器获取到所述服务器将要推送的初始化片段和媒体片段序列的信息：HTTP/2 中的推送告知方式和 WebSocket 中的推送内容告知方式；所述客户端通过如下任意一种或多种方式发送所述 MPD 请求消息：HTTP/2 请求报文的扩展首部的键值对形式和 WebSocket 的帧中的 JSON 键值对形式。

在又一种可选的实施例中，所述请求信息还包括：至少一个请求参数的情况下，所述预测的片段消息包括请求参数所请求的片段的属性；所述客户端通过如下任意一种或多种方式从所述媒体服务器获取到所述预测的片段消息：HTTP/2 中的推送告知方式和 WebSocket 中的推送内容告知方式；所述客户端通过如下任意一种或多种方式发送所述请求消息：HTTP/2 请求报文的扩展首部的键值对形式和 WebSocket 的帧中的 JSON 键值对形式。

可选地，根据本申请上述实施例，所述客户端通过 MPD 请求消息向所述媒体服务器发送所述快速启动类型推送指令。

在一种可选的实施例中，向媒体服务器发送快速启动类型推送指令包括：通过 MPD 请求消息向所述媒体服务器发送所述快速启动类型推送指令，其中，所述 MPD 请求消息用于请求所述媒体服务器发送 MPD 响应消息，所述 MPD 响应消息至少应包含 MPD 数据和快速启动类型推送确认指令。

可选地，根据本申请上述实施例，所述客户端获取所述快速启动类型推送确认指令包括：所述客户端接收所述快速启动类型推送确认指令包括：所述客户端接收所述媒体服务器返回的 MPD 响应消息，所述 MPD 响应消息至少包含 MPD 数据，以及所述快速启动类型推送确认指令。

在一种可选的实施例中，通过获取并解析 MPD 响应消息获取所述快速启动类型推送确认指令，所述 MPD 响应消息至少包含 MPD 数据和快速启动类型推送确认指令。

可选地，根据本申请上述实施例，所述客户端根据所述快速启动类型推送确认指令获取所述服务器将要推送的初始化片段和/或媒体片段序列的信息，包括：

步骤 S1061，所述快速启动类型推送确认指令中携带 URL 模板类型的推送参数。

步骤 S1063，所述客户端解析 URL 模板推送参数，获取所述媒体服务器将要推送的初始化片段序列和/或媒体片段序列的信息。

可选地，根据本申请上述实施例，所述客户端根据所述快速启动类型推送确认指令获取所述服务器将要推送的初始化片段和/或媒体片段序列的信息，包括：

步骤 S1065，所述快速启动类型推送确认指令中携带数值类型的推送参数。

步骤 S1067，所述客户端解析数值推送参数 K，并接收后续的 K 个推送承诺 PUSH\_PROMISE 帧，获取所述媒体服务器将要推送的初始化片段序列和/或媒体片段序列的信息。

可选地，根据本申请上述实施例，所述客户端根据所述快速启动类型推送确认指令获取所述服务器将要推送的初始化片段和/或媒体片段序列的信息，包括：

步骤 S1069，所述快速启动类型推送确认指令中携带 URL 列表类型的推送参数；

步骤 S1071，所述客户端解析 URL 列表推送参数，获取所述媒体服务器将要推送的初始化片段序列和/或媒体片段序列的信息。

在一种可选的实施例中，所述客户端根据所述快速启动类型推送确认指令获取所述服务器将要推送的初始化片段和媒体片段序列的信息的步骤包括：

在所述快速启动类型推送指令的推送参数 PUSH\_PARAMS 为 URL 模板 URL\_TEMPLATE 类型的情况下，所述客户端解析 URL 模板，获取媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息。

在所述快速启动类型推送指令的推送参数 PUSH\_PARAMS 为数值 NUMBER 类型的情况下，所述客户端通过解析数值 K 和后续的 K 个推送承诺 PUSH\_PROMISE 帧，获取媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息。

在所述快速启动类型推送指令的推送参数 PUSH\_PARAMS 为 URL 列表 URL\_LIST 类型的情况下，所述客户端解析 URL 列表，获取媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息。

可选地，根据本申请上述实施例，上述方法还包括：

步骤 S108，所述客户端解析所述 MPD 数据，确定需要所述媒体服务器推送的初始化片段和/或媒体片段序列的信息。

步骤 S1010，若所述获取的媒体服务器将要推送的初始化片段和/或媒体片段序列的信息等同于所述确定的需要媒体服务器推送的初始化片段和/或媒体片段序列的信息，所述客户端接收所述媒体服务器推送的初始化片段序列和/或媒体片段序列。

可选地，根据本申请上述实施例，上述方法还包括：

步骤 S1012，所述客户端解析所述 MPD 数据，确定需要所述媒体服务器推送的初始化片段和/或媒体片段序列的信息。

步骤所述客户端向所述媒体服务器发送获取所述确定需要所述媒体服务器推送的初始化片段和/或媒体片段序列的请求。

所述客户端向所述媒体服务器发送获取所述确定需要所述媒体服务器推送的初始化片段和/或媒体片段序列的请求。

所述客户端向所述媒体服务器发送获取所述确定需要所述媒体服务器推送的初始化片段和/或媒体片段序列的请求。

S1014, 若所述获取的媒体服务器将要推送的初始化片段和/或媒体片段序列的信息不同于所述确定的需要媒体服务器推送的初始化片段和/或媒体片段序列的信息, 所述客户端取消所述媒体服务器推送的初始化片段序列和/或媒体片段序列。

可选地, 在本申请上述实施例中, 上述方法还包括所述客户端向所述媒体服务器发送获取所述确定需要所述媒体服务器推送的初始化片段和/或媒体片段序列的请求。

在一种可选的实施例中, 本申请还提供了如图 5 所示的一种流媒体快速启动方法, 该方法包括:

步骤 161, 根据客户端是否需要快速启动和呈现描述 MPD, 生成 push-fast-start directive;

步骤 162, 向媒体服务器发送 MPD 片段请求消息, 其中, 所述 MPD 片段请求下拍戏由上述 push-fast-start directive 与 MPD 请求消息指令组成;

步骤 163, 从所述媒体服务器接收响应 MPD 片段请求消息的 MPD 片段, 并解析响应消息头部信息中 push-fast-start directive 中的参数;

步骤 164, 分析判断所需内容是否命中。

在一种可选的实施例中, 在所述快速启动类型推送指令的推送参数 PUSH\_PARAMS 为数值 NUMBER 类型的情况下, 所述客户端通过解析数值 K 和后续的 K 个推送承诺 PUSH\_PROMISE 帧, 获取媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息, 包括:

步骤 S1081, 所述客户端从所述快速启动类型推送指令中解析得到所述参数值 K, 并接收所述媒体服务器推送的与所述参数值 K 对应的 K 个片段消息。

步骤 S1083, 所述客户端解析所述 MPD 数据, 得到所述客户端需要的媒体内容的媒体信息。

步骤 S1085, 所述客户端检测所述 K 个片段消息中是否命中所述媒体信息, 所述片段消息为 URL 地址。

步骤 S1087, 如果在所述 K 个片段消息中命中所述媒体消息成功, 所述客户端获取所述媒体服务器将要推送的初始化片段和媒体片段序列的信息。

在一种可选的实施例中, 在所述推送确认指令中携带所述媒体服务器预测的参数值 K 的情况下, 所述客户端获取到预测的片段消息可以包括如下步骤:

所述客户端从所述推送确认指令中解析得到所述参数值 K, 并接收所述媒体服务器推送的与所述参数值 K 对应的 K 个片段消息。

在一种可选的实施例中, 根据所述预测的片段消息和所述 MPD 数据, 获取所述客户端需要的媒体内容, 包括,

所述客户端解析所述 MPD 数据，得到所述客户端需要的媒体内容的媒体信息。

所述客户端检测所述 K 个片段消息中是否命中所述媒体信息，所述片段消息为 URL 地址。

如果在所述 K 个片段消息中命中所述媒体消息成功，所述客户端获取所述媒体服务器推送的与所述预测的片段消息所对应的片段序列，其中，所述片段序列为所述客户端需要的媒体内容。

可选地，根据本申请上述实施例提供的方法，如果在所述 K 个片段消息中命中所述媒体消息失败，所述客户端发起获取所述需要的媒体内容的请求，或取消所述媒体服务器推送媒体内容的会话。

提供的带参数 K 的快速启动推送确认指令参数格式，可以定义为：**fast\_start** 指示推送确认指令，由参数 K 指定推送确认指令的类型，由 K 值组成带参数 K 的快速启动确认指令。具体示例结合如图 5B 至图 5C 所示。

在一种可选的实施例中，所述 **push-fast-start acknowledge** 参数可以由快速启动信息 (**fast-start**) 和一个参数 K 组成，除此以外，媒体服务器在推送预测片段序列的同时会先推送片段序列的消息，如 URL 等信息（该事先推送的片段序列信息的方式可以为 HTTP/2 中的 **push promise** 等方式，所述 **push promise** 指的是由媒体服务器发送，用来通知媒体客户端要推送的内容信息，比如 URL 信息等），给媒体客户端，以便于媒体客户端通过 **push-fast-start acknowledge** 参数 K 和推送片段序列的信息来尽早决策媒体服务器端是否命中失败 (Miss)。

本公开实施例提供了一种流媒体快速启动方法，图 6 是根据本公开实施例的一种流媒体快速启动方法的流程图，图 6 是根据本公开实施例的一种可以由用户端设备(如，媒体客户端 110)实现的流媒体快速启动方法 200 的流程示意图，如图 6 所示，该流程包括以下步骤：

步骤 201，根据客户端是否需要快速启动和呈现描述 MPD，生成 **push-fast-start directive**，其中，所述 **push-fast-start directive** 可以不包含任何参数，也可以包含一些表示客户端偏好的参数，比如加入客户端偏好的码率，偏好的媒体片段类型，偏好的媒体片段个数等或这些的组合等；

步骤 202，向媒体服务器发送 MPD 片段请求消息，其中，所述 MPD 片段请求消息由上述 **push-fast-start directive** 与 MPD 请求消息指令组成；

步骤 203，从所述媒体服务器接收响应 MPD 片段请求消息的 MPD 片段，并解析响应消息头部信息中 **push-fast-start directive** 中的参数值 K；

步骤 204，从所述媒体服务器接收服务器推送的 K 个片段的片段消息，判断所需内容是否命中，命中的话等待获取内容，命中失败的话，可以立即发送对应所需消息的请求，也可以取消 (Cancel) 媒体服务器的推送，直接开始请求所需片段序列。

所述步骤 202 中，所述片段请求消息具体可通过且不限于以下形式发送：**HTTP/2** 请求报

文的扩展首部的键值对形式、WebSocket 的帧中的 JSON 键值对形式等。

所述步骤 204 中，所述参数 K 对应的片段序列的消息被媒体服务器同步推送，所述参数 K 对应的片段序列被媒体服务器同步推送（或者其他更加快速，公平的方法推送）。

基于以上步骤，媒体客户端通过生成所述 `push-fast-start directive`，即可告诉媒体服务器，客户端期待媒体服务器推送片段序列，以快速启动；通过使用从所述媒体服务器接收服务器推送的 K 个片段的片段消息，减少了客户端盲目等待的时间，降低了启动延迟。

所述方法 200 为示例性实施例，可以对所述方法 200 进行替代性修改，且可以根据需要合并和/或附加步骤。

下文是一个示例性实施例，用以说明在此介绍的流媒体快速启动方法和参数值 K，及其在一些用例中的潜在应用。

图 7A 示出了示例性 MPD 400。该 MPD 取自 2016 ICME Grand Challenge 竞赛组委会提供的示例文件（MPD 文件亦可以是其他标准类型的 DASH MPD 文件）。

图 7B 示出了示例性片段请求消息 410。片段请求消息 410 包括了 `mpd_uri` 411 和 `fast_start` 推送指令 412，不含参数。

图 7B-1 示出了示例性片段请求消息 410-1。片段请求消息 410-1 包括了 `mpd_uri` 411-1 和 `fast_start` 推送指令 412-1，该指令由快速启动信息(`fast-start`)和推送参数组成。所述推送参数有多种，可以为 `video = V`、`audio = A`、`init-only`、`biterate=B` 或者它们相互组合（`video`、`audio`、`biterate` 数值或者范围亦可不指定）。

图 7C 示出了示例性片段响应消息 420。片段响应消息 420 包括了 `mpd_content` 421 和 `fast_start` 推送确认指令 412，包含参数值 K，亦即媒体服务器欲推送片段的个数。

图 7D 示出了媒体客户端 110 与媒体服务器 120 的示例性交互流程。交互流程起始于步骤 431，媒体客户端 110 通过 FDH 访问引擎 113 向所述媒体服务器 120 发送 MPD 请求消息，请求 MPD400，并告诉媒体服务器需要快速启动。在步骤 432 中，媒体服务器 120 接收到 MPD 请求消息后，通过 FDH 服务模块 121 获取预测片段的数目以及具体内容，向媒体客户端 110 发送 MPD 响应消息，所述 MPD 响应消息包含 MPD400 与 `push-fast-start acknowledge`422。在步骤 433 中，媒体服务器 120 服务模块 121 根据本地预测和 MPD 信息生成预测片段信息，并生成要推送的片段序列。同时向媒体客户端并行发送要推送片段的信息（URL 等信息，不包含具体数据）。在步骤 434 中，所述媒体客户端 110 通过 FDH-DASH 控制引擎 111 解析，所需片段是否在服务器推送的片段消息中，如果在等待获取文件内容，否则，立马请求所需片段。媒体服务器在步骤 435 中，通过服务模块 121 向所述媒体客户端 110 并行发送片段请求消息。在步骤 436 中，所述媒体客户端 110 正常发送片段请求。步骤 434 和步骤 435 可以同时进行，以便于媒体客户端尽早决策。

可选地，根据本申请上述实施例，在所述推送确认指令中携带所述媒体服务器预测的推

送片段列表的情况下，所述推送片段列表为从所述推送确认指令这种解析得到的所述预测的片段消息，其中，根据所述预测的片段消息和所述 MPD 数据，获取所述客户端需要的媒体内容，包括，

步骤 S1067，所述客户端解析所述 MPD 数据，得到所述客户端需要的媒体内容的媒体信息；

步骤 S1069，所述客户端检测所述推送片段列表中是否命中所述媒体信息；

步骤 S1071，如果在所述推送片段列表中命中所述媒体消息成功，所述客户端获取所述媒体服务器推送的与所述推送片段列表所对应的片段序列，其中，所述片段序列为所述客户端需要的媒体内容。

可选地，根据本申请上述实施例，如果在所述 K 个片段消息中命中所述媒体消息失败，所述客户端发起获取所述需要的媒体内容的请求并取消所述媒体服务器推送媒体内容的会话，或所述客户端发起获取所述需要的媒体内容的请求并存储所述媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息。

在一种可选的实施例中，如果在所述推送片段列表中命中所述媒体消息失败，所述客户端发起获取所述需要的媒体内容的请求，或取消所述媒体服务器推送媒体内容的会话。

可选地，根据本申请上述实施例，所述推送片段列表为所述媒体服务器预测的片段序列的 URL 组成。

本公开介绍一种新的 URL\_LIST 快速启动推送确认指令参数格式，可以定义为：

ATTRIBUTES = ATTRIBUTE\_LIST / ATTRIBUTE\_ITEM

ATTRIBUTE\_LIST = ATTRIBUTE\_ITEM";"ATTRIBUTE\_LIST/ATTRIBUTE\_ITEM

ATTRIBUTE\_ITEM 可以为初始化片段，媒体类型，语言，比特率等，值可以为数目，码率，范围等。

fast\_start 指示推送确认指令，由参数 segment-list 指定推送确认指令的类型，然后跟上所有片段的 URL，URL 由符号";"分割组成 URL\_LIST 快速启动确认指令。具体示例如图 9B-C 所示。

可选地，根据本申请上述实施例，在所述快速启动类型推送指令的推送参数 PUSH\_PARAMS 为 URL 列表 URL\_LIST 类型的情况下，所述客户端解析 URL 列表，获取媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息，包括，

步骤 S1091，所述客户端解析所述 MPD 数据，得到所述客户端需要的媒体内容的媒体信息。

步骤 S1093，所述客户端检测所述媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息是否命中所述媒体信息。

步骤 S1095, 如果在所述媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息命中所述媒体消息成功, 获取媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息。

在一种可选的实施例中, 所述 `push-fast-start acknowledge` 参数可以由快速启动信息 (`fast-start`) 和即将推送片段列表 (`URL_LIST`) 组成, 所述 `URL_LIST` 由一系列媒体服务器预测的片段序列的 `URL` 组成, 这些 `URL` 列表可以由分号 “;” 分割 (或者其他表现形式), 交给媒体客户端解析。媒体客户端可以通过 `MPD` 请求响应消息头中的 `push-fast-start directive` 参数, 解析获取服务器推送片段的多少, 以及具体推送的内容, 通过这种方式, 媒体客户端可以尽早的做出决策。

本公开实施例提供了一种流媒体快速启动方法, 图 8 是根据本公开实施例的一种流媒体快速启动方法的流程图, 图 8 是根据本公开实施例的一种可以由用户端设备 (如, 媒体客户端 110) 实现的流媒体快速启动方法 500 的流程示意图, 如图 8 所示, 该流程包括以下步骤:

步骤 501, 根据客户端是否需要快速启动和呈现描述 `MPD`, 生成 `push-fast-start directive`, 其中, 所述 `push-fast-start directive` 可以不包含任何参数, 也可以包含一些表示客户端偏好的参数, 比如加入客户端偏好的码率, 偏好的媒体片段类型, 偏好的媒体片段个数等或这些的组合等;

步骤 502, 向媒体服务器发送 `MPD` 片段请求消息, 其中, 所述 `MPD` 片段请求消息由上述 `push-fast-start directive` 与 `MPD` 请求消息指令组成;

步骤 503, 从所述媒体服务器接收响应 `MPD` 片段请求消息的 `MPD` 片段, 并解析响应消息头部信息中 `push-fast-start directive` 中的 `URL_LIST` 参数列表。判断媒体客户端需要的片段是否存在于参数列表, 存在的话则等待获取媒体服务器推送的内容, 不存在的话 (Miss), 直接请求所需片段序列。

所述步骤 502 中, 所述片段请求消息具体可通过且不限于以下形式发送: `HTTP/2` 请求报文的扩展首部的键值对形式、`WebSocket` 的帧中的 `JSON` 键值对形式等。

所述步骤 504 中, 所述参数 `URL_LIST` 参数列表由媒体服务器生成, 该列表可以根据多数媒体客户端的命中情况来预测, 或者实现指定配置文件, 或者通过结合媒体客户端 `push-fast-start directive` 中的参数情况等。

基于以上步骤, 媒体客户端通过生成所述 `push-fast-start directive`, 即可告诉媒体服务器, 客户端期待媒体服务器推送片段序列, 以快速启动; 通过使用从所述媒体服务器接收服务器推送的 `push-fast-start acknowledge` 中的 `URL_LIST` 参数列表, 减少了客户端盲目等待的时间, 降低了启动延迟。所述方法 500 为示例性实施例, 可以对所述方法 500 进行替代性修改, 且可以根据需要合并和/或附加步骤。

下文是一个示例性实施例, 用以说明在此介绍的流媒体快速启动方法和参数 `URL_LIST`, 及其在一些用例中的潜在应用。

图 9A 示出了示例性 MPD 700。该 MPD 取自 2016 ICME Grand Challenge 竞赛组委会提供的示例文件（MPD 文件亦可以是其他标准类型的 DASH MPD 文件）。

图 9B 示出了示例性片段请求消息 710。片段请求消息 710 包括了 mpd\_uri 711 和 fast\_start 推送指令 712，不含参数。

图 9B-1 示出了示例性片段请求消息 710-1。片段请求消息 710-1 包括了 mpd\_uri 711-1 和 fast\_start 推送指令 712-1，该指令由快速启动信息(fast-start)和推送参数组成。所述推送参数有多种，可以为 video = V、audio = A、init-only、biterate=B 或者它们相互组合（video、audio、biterate 数值或者范围亦可不指定）。

图 9C 示出了示例性片段响应消息 720。片段响应消息 720 包括了 mpd\_content 721 和 fast\_start 推送确认指令 712，包含参数 URL\_LIST，亦即媒体服务器欲推送片段的信息列表（URL 等，不包含具体数据）。

图 9D 示出了媒体客户端 110 与媒体服务器 120 的示例性交互流程。交互流程起始于步骤 731，媒体客户端 110 通过 FDH 访问引擎 113 向所述媒体服务器 120 发送 MPD 请求消息，请求 MPD700，并告诉媒体服务器需要快速启动。在步骤 732 中，媒体服务器 120 接收到 MPD 请求消息后，通过 FDH 服务模块 121 获取预测片段的数目以及具体内容，向媒体客户端 110 发送 MPD 响应消息，所述 MPD 响应消息包含 MPD400 与 push-fast-start acknowledge722。在步骤 733 中，所述媒体客户端 110 通过 FDH-DASH 控制引擎 111 解析，所需片段是否在服务器推送的片段列表中，如果在等待获取文件内容，否则，立马请求所需片段。媒体服务器在步骤 734 中，通过服务模块 121 向所述媒体客户端 110 并行发送片段请求消息。在步骤 735 中，所述媒体客户端 110 正常发送片段请求。步骤 733 和步骤 734 可以同时进行，以便于媒体客户端尽早决策。

可选地，根据本申请上述实施例，在所述快速启动类型推送指令的推送参数 PUSH\_PARAMS 为 URL 模板 URL\_TEMPLATE 类型的情况下，所述客户端解析 URL 模板，获取媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息，包括：

步骤 S1073，所述客户端解析所述 MPD 数据，得到所述客户端需要的媒体内容的媒体信息。

步骤 S1075，所述客户端检测所述媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息是否命中所述媒体信息。

步骤 S1077，如果在所述媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息命中所述媒体消息成功，获取媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息。

在一种可选的实施例中，在所述推送确认指令中携带所述媒体服务器预测的推送片段信息的情况下，所述推送片段信息为从所述推送确认指令这种解析得到的所述预测的片段消息，其中，根据所述预测的片段消息和所述 MPD 数据，获取所述客户端需要的媒体内容，包括，

所述客户端解析所述 MPD 数据，得到所述客户端需要的媒体内容的媒体信息；

所述客户端检测所述推送片段消息中是否命中所述媒体信息；

如果在所述推送片段消息中命中所述媒体消息成功，所述客户端获取所述媒体服务器推送的与所述推送片段消息所对应的片段序列，其中，所述片段序列为所述客户端需要的媒体内容。

可选地，根据本申请上述实施例，如果在所述媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息命中所述媒体消息失败，所述客户端发起获取所述需要的媒体内容的请求并取消所述媒体服务器推送媒体内容的会话，或所述客户端发起获取所述需要的媒体内容的请求并存储所述媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息。

在一种可选的实施例中，如果在所述推送片段消息中命中所述媒体消息失败，所述客户端发起获取所述需要的媒体内容的请求，或取消所述媒体服务器推送媒体内容的会话。

可选地，根据本申请上述实施例，所述推送片段列消息为所述媒体服务器预测的一个或多个带参数化遍历的 URL 模板和压缩后的参数列表组成。

本公开介绍一种新的 URL\_TEMPLATE 快速启动推送确认指令参数格式，可以定义为：

ATTRIBUTES = ATTRIBUTE\_LIST / ATTRIBUTE\_ITEM

ATTRIBUTE\_LIST = "ATTRIBUTE\_ITEM";"ATTRIBUTE\_LIST/ATTRIBUTE\_ITEM

ATTRIBUTE\_ITEM 可以为初始化片段，媒体类型，语言，比特率等，值可以为数目，码率，范围等。

fast\_start 指示推送确认指令，由参数 push-template 指定推送确认指令的类型，然后跟上标准 push-template 以及上述描述的一些参数组成 URL\_TEMPLATE 快速启动确认指令。具体示例如图 10B 至图 10C 所示。

在一实施例中，所述 push-fast-start acknowledge 参数可以由快速启动信息(fast-start)和即将推送片段信息 (URL\_TEMPLATE)组成，所述 URL\_TEMPLATE 由包含一个或多个待参数化变量的 URL 模板和压缩后的参数列表组成，所述 URL 模板对应于 IETF RFC 6570 URI template Level 1，变量包含于“{}”，表达式类型为字符串扩展模式（详见 FDH URL\_TEMPLATE 定义），交给媒体客户端解析。媒体客户端可以通过 MPD 请求响应消息头中的 push-fast-start directive 参数，解析获取服务器推送片段的多少，以及具体推送的内容，通过这种方式，媒体客户端可以尽早的做出决策。

本公开实施例提供了一种流媒体快速启动方法，图 11 是根据本公开实施例的一种流媒体快速启动方法的流程图，图 11 是根据本公开实施例的一种可以由用户端设备(如，媒体客户端 110)实现的流媒体快速启动方法 800 的流程示意图，如图 11 所示，该流程包括以下步骤：

步骤 801，根据客户端是否需要快速启动和呈现描述 MPD，生成 push-fast-start directive，

其中，所述 `push-fast-start directive` 可以不包含任何参数，也可以包含一些表示客户端偏好的参数，比如加入客户端偏好的码率，偏好的媒体片段类型，偏好的媒体片段个数等或这些的组合等；

步骤 802，向媒体服务器发送 MPD 片段请求消息，其中，所述 MPD 片段请求消息由上述 `push-fast-start directive` 与 MPD 请求消息指令组成；

步骤 803，从所述媒体服务器接收响应 MPD 片段请求消息的 MPD 片段，并解析响应消息头部信息中 `push-fast-start directive` 中的 `URL_TEMPLATE` 参数。判断媒体客户端需要的片段是否存在于参数列表，存在的话则等待获取媒体服务器推送的内容，不存在的话(Miss)，直接请求所需片段序列。

所述步骤 802 中，所述片段请求消息具体可通过且不限于以下形式发送：`HTTP/2` 请求报文的扩展首部的键值对形式、`WebSocket` 的帧中的 `JSON` 键值对形式等。

所述步骤 804 中，所述参数 `URL_TEMPLATE` 参数列表由媒体服务器生成，该列表可以根据多数媒体客户端的命中情况来预测，或者实现指定配置文件等。

基于以上步骤，媒体客户端通过生成所述 `push-fast-start directive`，即可告诉媒体服务器，客户端期待媒体服务器推送片段序列，以快速启动；通过使用从所述媒体服务器接收服务器推送的 `push-fast-start acknowledge` 中的 `URL_TEMPLATE` 参数列表，减少了客户端盲目等待的时间，降低了启动延迟，并且对于 `push-fast-start acknowledge` 进行了更加合理的压缩传输。所述方法 800 为示例性实施例，可以对所述方法 800 进行替代性修改，且可以根据需要合并和/或附加步骤。

下文是一个示例性实施例，用以说明在此介绍的流媒体快速启动方法和参数 `URL_TEMPLATE`，及其在一些用例中的潜在应用。

图 10A 示出了示例性 MPD 1000。该 MPD 取自 2016 ICME Grand Challenge 竞赛组委会提供的示例文件（MPD 文件亦可以是其他标准类型的 `DASH MPD` 文件）。

图 10B 示出了示例性片段请求消息 1010。片段请求消息 1010 包括了 `mpd_uri` 1011 和 `fast_start` 推送指令 1012，不含参数。

图 10B-1 示出了示例性片段请求消息 1010-1。片段请求消息 1010-1 包括了 `mpd_uri` 1011-1 和 `fast_start` 推送指令 1012-1，该指令由快速启动信息(`fast-start`)和推送参数组成。所述推送参数有多种，可以为 `video = V`、`audio = A`、`init-only`、`biterate=B` 或者它们相互组合（`video`、`audio`、`biterate` 数值或者范围亦可不指定）。

图 10C 示出了示例性片段响应消息 1020。片段响应消息 1020 包括了 `mpd_content` 1021 和 `fast_start` 推送确认指令 1012，包含参数 `URL_TEMPLATE`，亦即媒体服务器欲推送片段的信息列表压缩形式（此处可以为 `FDH-DASH` 标准中的定义格式等）。

图 10D 示出了媒体客户端 110 与媒体服务器 120 的示例性交互流程。交互流程起始于步

骤 1031, 媒体客户端 110 通过 FDH 访问引擎 113 向所述媒体服务器 120 发送 MPD 请求消息, 请求 MPD1000, 并告诉媒体服务器需要快速启动。在步骤 1032 中, 媒体服务器 120 接收到 MPD 请求消息后, 通过 FDH 服务模块 121 获取预测片段的数目以及具体内容, 向媒体客户端 110 发送 MPD 响应消息, 所述 MPD 响应消息包含 MPD400 与 push-fast-start acknowledge 1022。在步骤 1033 中, 所述媒体客户端 110 通过 FDH-DASH 控制引擎 111 解析, 所需片段是否在服务器推送的片段列表中, 如果在等待获取文件内容, 否则, 立马请求所需片段。媒体服务器在步骤 1034 中, 通过服务模块 121 向所述媒体客户端 110 并行发送片段请求消息。在步骤 1035 中, 所述媒体客户端 110 正常发送片段请求。步骤 1033 和步骤 1034 可以同时进行, 以便于媒体客户端尽早决策。

图 13 是计算机系统或网络设备 1200 的实施例的示意图。所述网络设备 1200 可以为任意合适的设备予以实现, 如, 网络服务器、媒体服务器(如, 媒体服务器 120)或用户端设备(如, 媒体客户端 110)等等。所述网络设备 1200 能够接收、处理和发送去向以及来自网络的片段请求消息/片段响应消息/片段等消息。所述网络设备 1200 可以包括与接收器 1211 连接的一个或多个入口端 1210, 其中, 所述接收器 1211 可用于通过 FDH 通道从其他网络节点接收 MPD 和/或片段。所述网络设备 1200 还可以包括与发送器 1231 连接的一个或多个出口端 1230, 其中, 所述发送器 1231 可以通过 FDH 通道向其他网络节点发送 MPD 和/或片段。所述网络设备 1200 还可以包括处理器 1220, 其中, 所述处理器 1220 与所述接收器 1211 链接, 设置为处理所述片段或者确定向那个网络节点发送所述片段。

所述处理器 1220 可以通过硬件、软件及其组合来实现。所述处理器 1220 可用一个或多个 CPU 芯片、可编程门阵列、专用集成电路和/或数字信号处理器予以实现。所述处理器 1220 可用于实现在此描述的任意功能模块或单元, 例如, FDH-DASH 控制引擎 111、FDH 访问引擎 113、媒体引擎 112、FDH 服务模块 121、解析引擎 122、媒体播放器 1221 或者本领域普通技术人员熟知的其他任何功能组件, 或者其任何组合。所述媒体播放器 1221 可以为在所述处理器 1220 上运行的应用程序, 设置为解码播放所述媒体片段。

所述网络设备 1200 还可以包括至少一个存储器 1222。所述存储器 1222 可设置为存储 MPD1223 和片段 1224。所述 MPD1223 和所述片段 1224 可以通过以上所述的各种形式进行存储和打包。例如, 所述 MPD1223 可以为所述 MPD300(或者其他在 MPEG-DASH 中定义的标准 MPD 文件)。如图 13 所示, 所述片段 1224 和所述 MPD1223 可以作为单独的文件进行存储, 也可以将其存储在单个文件中。

在实践中, 可能存在所述网络设备 1200 处理的双向流量, 因此, 一些端口可既接收也发送报文。在这种情况下, 所述入口端 1210 和出口端 1230 可以是共设在一起的或者可以认识始于收发器连接的不同端口的不同功能。所述处理器 1220、所述存储器 1222、所述接收器 1211 和所述发送器 1231 中的一个或多个也可以设置为至少部分实现或支持上文描述的任意方法和实现方式。

在本实施例中还提供了另一种流媒体快速启动方法, 图 14 是根据本公开实施例的一种流媒体快速启动方法的流程图, 如图 14 所示, 该流程包括如下步骤:

步骤 S1401, 媒体服务器接收客户端发送的快速启动类型推送指令, 其中, 所述快速启动类型推送指令用于指示所述媒体服务器推送初始化片段序列和/或媒体片段序列。

步骤 S1403, 所述媒体服务器根据将要推送的初始化片段和/或媒体片段序列, 生成快速启动类型推送确认指令。

步骤 S1405, 所述媒体服务器向所述客户端发送所述快速启动类型推送确认指令。

在一种可选的实施例中, 上述流媒体快速启动方法的步骤可以包括:

媒体服务器获取快速启动类型推送指令 `push-fast-start directive`; 所述媒体服务器生成快速启动类型推送确认指令 `push-fast-start acknowledge`, 其中, 所述快速启动类型推送确认指令用于通知客户端将要推送的推送初始化片段序列和/或媒体片段序列; 向所述客户端发送所述快速启动类型推送确认指令, 向所述客户端推送初始化片段序列和/或媒体片段序列。

可选地, 在本申请上述实施例中, 所述媒体服务器接收所述客户端发送的 MPD 请求消息, 所述 MPD 请求消息中包含所述快速启动类型推送指令。

可选地, 在本申请上述实施例中, 媒体服务器接收快速启动类型推送指令包括: 所述媒体服务器接收所述客户端发送的 MPD 请求消息, 所述 MPD 请求消息中包含所述快速启动类型推送指令。

可选地, 在本申请上述实施例中, 所述媒体服务器生成快速启动类型推送确认指令, 包括:

步骤 S14031, 所述媒体服务器根据将要推送的初始化片段序列和/或媒体片段序列, 生成 URL 模板类型的推送参数。

步骤 S14033, 所述快速启动类型推送确认指令中携带所述 URL 模板类型的推送参数。

可选地, 在本申请上述实施例中, 所述媒体服务器生成快速启动类型推送确认指令, 包括:

步骤 S14035, 所述媒体服务器根据将要推送的初始化片段序列和/或媒体片段序列, 生成数值类型的推送参数。

步骤 S14037, 所述快速启动类型推送确认指令中携带所述数值类型的推送参数。

可选地, 在本申请上述实施例中, 所述媒体服务器生成快速启动类型推送确认指令, 包括:

步骤 S14039, 所述媒体服务器根据将要推送的初始化片段序列和/或媒体片段序列, 生成 URL 列表类型的推送参数。

步骤 S14040, 所述快速启动类型推送确认指令中携带所述 URL 列表类型的推送参数。

可选地, 在本申请上述实施例中, 发送所述快速启动类型推送确认指令包括:

所述媒体服务器通过 MPD 响应消息向所述客户端发送所述快速启动类型推送确认指令，其中，所述 MPD 响应消息至少包含 MPD 数据，以及所述快速启动类型推送确认指令。

可选地，在本申请上述实施例中，所述媒体服务器生成快速启动类型推送确认指令，包括：

所述媒体服务器根据所述客户端的预定数量和/或所述客户端的历史访问结果，确定所述快速启动类型推送确认指令中携带的推送参数。

可选地，在本申请上述实施例中，所述媒体服务器向所述客户端发送所述快速启动类型推送确认指令之后，上述方法还包括：

所述媒体服务器向所述客户端推送所述快速启动类型推送确认指令指示的片段序列初始化片段和/或媒体片段序列。

可选地，在本申请上述实施例中，所述 MPD 请求消息还包括：所述 MPD 请求消息中携带的快速启动推送指令包含偏好参数信息，所述偏好参数包括如下任意一种或多种属性：分辨率范围、码率范围、媒体属性和语言。

在一种可选的实施例中，所述 MPD 请求消息还包括：至少一个请求参数的情况下，所述初始化片段和媒体片段序列的信息包括请求参数所请求的片段的属性。

可选地，在本申请上述实施例中，所述媒体服务器接收所述客户端发送的 MPD 请求消息，以及所述快速启动类型推送指令，进一步包括：

通过 HTTP/2 报文方式接收所述 MPD 请求消息，其中，所述 HTTP/2 报文的 Accept-Push-Policy 首部携带快速启动类型推送指令；或者，

通过 WebSocket 中 DASH 子协议的 get\_mpd 消息接收所述 MPD 请求消息，其中，所述 get\_mpd 消息的 JSON 键值对携带快速启动类型推送确认指令。

可选地，在本申请上述实施例中，所述媒体服务器向所述客户端发送 MPD 响应消息，以及所述快速启动类型推送确认指令，进一步包括：

通过 HTTP/2 报文方式发送所述 MPD 响应消息，其中，所述 HTTP/2 报文的 Accept-Push-Policy 首部携带快速启动类型推送确认指令；或者，

通过 WebSocket 中 DASH 子协议的新新\_mpd 消息发送所述 MPD 响应消息，其中，所述 new\_mpd 消息的 JSON 键值对携带快速启动类型推送确认指令。

在一种可选的实施例中，所述媒体服务器通过如下任意一种或多种方式推送初始化片段和媒体片段序列的信息：HTTP/2 的 MPD 响应报文方式和 WebSocket 中 DASH 子协议的新新\_mpd 消息，其中，HTTP/2 的 MPD 响应报文通过 Push-Policy 首部携带快速启动类型推送确认指令；WebSockets 中 DASH 子协议的新新\_mpd 消息通过 JSON 键值对形式携带快速启动类型推送确认指令。

在一种可选的实施例中，媒体服务器获取快速启动类型推送指令 `push-fast-start directive` 包括：通过获取并解析 MPD 请求消息获取所述快速启动类型推送指令，所述 MPD 请求消息至少包含 MPD URI 和快速启动类型推送指令；所述媒体服务器生成快速启动类型推送确认指令 `push-fast-start acknowledge`。

在一种可选的实施例中，所述媒体服务器根据将要推送的初始化片段序列和/或媒体片段序列，生成快速启动类型推送确认指令，其中，所述快速启动类型推送确认指令的推送参数可为 URL 模板 URL\_TEMPLATE 类型，和/或 数值 NUMBER 类型，和/或 URL 列表 URL\_LIST 类型。

在一种可选的实施例中，向所述客户端发送所述快速启动类型推送确认指令包括：通过 MPD 响应消息向所述客户端发送所述快速启动类型推送确认指令，其中，所述 MPD 响应消息用于响应所述客户端发送 MPD 请求消息，所述 MPD 响应消息至少包含 MPD 数据和所述快速启动类型推送确认指令。

在一种可选的实施例中，上述流媒体快速启动方法包括：

媒体服务器接收客户端发送的用于获取媒体呈现描述文件 MPD 的请求消息，其中，请求信息至少包括：快速启动推送指令和 MPD 请求指令；

媒体服务器根据请求消息生成响应信息，其中，响应消息包括：确定客户端需要快速启动的推送确认指令、响应 MPD 请求指令得到的 MPD 数据；

在媒体服务器生成预测的片段消息之后，媒体服务器将预测的片段消息和响应信息推送至客户端，使得客户端根据预测的片段消息和 MPD 数据，获取客户端需要的媒体内容，其中，预测的片段消息为媒体服务器主动推送的与推送确认指令对应的片段消息。

图 15 是根据本申请实施例的一种可选的流媒体快速启动方法，结合图 15 所示，该方法包括：

步骤 171，从所述媒体客户端接收 MPD 片段请求消息，其中，所述片段请求消息包含 MPD 请求消息和 `push-fast-start directive`。

步骤 172，根据服务器的预测，生成 `push-fast-start acknowledge` 参数，向所述媒体客户端发送 MPD 片段相应消息，其中，所述片段相应消息包含 `push-fast-start directive` 和 MPD 片段。

步骤 173，根据所述 `push-fast-start acknowledge` 参数，确定所述媒体客户端可能期望的片段序列。

步骤 174，向所述媒体客户端推送预测片段序列。

下文是一个示例性实施例，用以说明在此介绍的流媒体快速启动方法和参数 URL\_LIST，及其在一些用例中的潜在应用。

图 9A 示出了示例性 MPD 700。该 MPD 取自 2016 ICME Grand Challenge 竞赛组委会提

供的示例文件（MPD 文件亦可以是其他标准类型的 DASH MPD 文件）。

图 9B 示出了示例性片段请求消息 710。片段请求消息 710 包括了 mpd\_uri 711 和 fast\_start 推送指令 712，不含参数。

图 9B-1 示出了示例性片段请求消息 710-1。片段请求消息 710-1 包括了 mpd\_uri 711-1 和 fast\_start 推送指令 712-1，该指令由快速启动信息(fast-start)和推送参数组成。所述推送参数有多种，可以为 video = V、audio = A、init-only、biterate=B 或者它们相互组合（video、audio、biterate 数值或者范围亦可不指定）。

图 9C 示出了示例性片段响应消息 720。片段响应消息 720 包括了 mpd\_content 721 和 fast\_start 推送确认指令 712，包含参数 URL\_LIST，亦即媒体服务器欲推送片段的信息列表（URL 等，不包含具体数据）。

图 9D 示出了媒体客户端 110 与媒体服务器 120 的示例性交互流程。交互流程起始于步骤 731，媒体客户端 110 通过 FDH 访问引擎 113 向所述媒体服务器 120 发送 MPD 请求消息，请求 MPD700，并告诉媒体服务器需要快速启动。在步骤 732 中，媒体服务器 120 接收到 MPD 请求消息后，通过 FDH 服务模块 121 获取预测片段的数目以及具体内容，向媒体客户端 110 发送 MPD 响应消息，所述 MPD 响应消息包含 MPD400 与 push-fast-start acknowledge722。在步骤 733 中，所述媒体客户端 110 通过 FDH-DASH 控制引擎 111 解析，所需片段是否在服务器推送的片段列表中，如果在等待获取文件内容，否则，立马请求所需片段。媒体服务器在步骤 734 中，通过服务模块 121 向所述媒体客户端 110 并行发送片段请求消息。在步骤 735 中，所述媒体客户端 110 正常发送片段请求。步骤 733 和步骤 734 可以同时进行，以便于媒体客户端尽早决策。

可选地，根据本申请上述实施例，在所述快速启动类型推送确认指令的推送参数为数值 NUMBER 类型的情况下，根据将要推送的初始化片段序列和/或媒体片段序列，生成快速启动类型推送确认指令，包括：

所述媒体服务器结合预定数量的媒体客户端和/或所述客户端的历史访问结果确定数值 K，并确定与所述数值 K 对应的 K 个片段消息。

所述媒体服务器将所述数值 K 加载至所述快速启动类型推送确认指令，

其中，所述媒体服务器将所述初始化片段序列和/或媒体片段序列推送至所述客户端包括：所述媒体服务器将所述数值 K 以及与所述数值 K 对应的 K 个片段消息推送至所述客户端。

可选地，在一种可选的实施例中，在所述推送确认指令中携带所述媒体服务器预测的参数值 K 的情况下，所述预测的片段消息为 K 个片段消息，其中，所述媒体服务器根据所述请求消息生成响应信息，包括：

所述媒体服务器结合预定数量的媒体客户端和/或所述客户端的历史访问结果确定预测的参数值 K，并确定与所述参数值 K 对应的 K 个片段消息；

所述媒体服务器将所述预测的参数值 K 加载至所述推送确认指令之后，得到所述响应信息；

其中，所述媒体服务器将所述预测的片段消息和所述响应信息推送至所述客户端包括：所述媒体服务器将所述参数值 K 以及与所述参数值 K 对应的 K 个片段消息推送至所述客户端。

本公开实施例中提供了一种流媒体快速启动方法，图 16 是根据本公开实施例的一种流媒体快速启动方法的流程图，图 16 是根据本公开实施例的一种可以由网络服务器(如，媒体服务器 120)实现的流媒体快速启动方法 300 的流程示意图，如图 16 所示，该流程包括如下步骤：

步骤 301，从所述媒体客户端接收 MPD 片段请求消息，其中，所述片段请求消息包含 MPD 请求信息和 push-fast-start directive；

步骤 302，根据服务器端的预测，生成 push-fast-start acknowledge 参数值 K，向所述媒体客户端发送 MPD 片段响应消息，其中，所述片段响应消息包含 push-fast-start acknowledge 和 MPD 片段；

步骤 303，根据所述 push-fast-start acknowledge 参数值 K，确定所述媒体客户端可能期望推送的 K 个片段序列；

步骤 304，向所述媒体客户端推送上述 K 个片段序列的片段消息和片段序列。

所述步骤 302 中，所述 push-fast-start acknowledge 参数值 K 可以根据服务器所有客户端预测结果来进行预测，如果大部分预测成功，适当降低阈值，如果大部分预测失败，适当提高阈值，否则不变（亦可以通过初始化指定且不限于所介绍的 K 值生成方法，或者结合媒体客户端 push-fast-start directive 中的参数决策等）。

所述步骤 303 中，所述 push-fast-start acknowledge 保证的参数值 K 个片段序列可以根据服务器所有客户端预测结果来进行预测（亦可以通过初始化指定且不限于所介绍的 K 个片段序列指定方法）。

所述步骤 304 中，所述 K 个片段序列的片段消息，用来通知客户端要推送的片段消息，比如片段 URL 等(或者可以通过其他告知方式，比如 HTTP/2 中的推送告知方式——push promise 和 WebSocket 中的推送内容告知方式等)。

基于以上步骤，媒体服务器只需根据所述 push-fast-start directive，在服务器指定推送个数 K 以及 K 个片段序列，仅需通过 push-fast-start acknowledge 和 K 个片段序列的消息，即可让客户端快速决策，减少了服务器驱动的盲目性，客户端驱动的不合理性，实现了 FDH-DASH 对媒体客户端的透明性，很好的达到了快速启动的目的。所述方法 300 为示例性实施例，可以根据需要进行修改。

下文是一个示例性实施例，用以说明在此介绍的流媒体快速启动方法和参数值 K，及其在一些用例中的潜在应用。

图 7A 示出了示例性 MPD 400。该 MPD 取自 2016 ICME Grand Challenge 竞赛组委会提供的示例文件（MPD 文件亦可以是其他标准类型的 DASH MPD 文件）。

图 7B 示出了示例性片段请求消息 410。片段请求消息 410 包括了 mpd\_uri 411 和 fast\_start 推送指令 412，不含参数。

图 7B-1 示出了示例性片段请求消息 410-1。片段请求消息 410-1 包括了 mpd\_uri 411-1 和 fast\_start 推送指令 412-1，该指令由快速启动信息(fast-start)和推送参数组成。所述推送参数有多种，可以为 video = V、audio = A、init-only、biterate=B 或者它们相互组合（video、audio、biterate 数值或者范围亦可不指定）。

图 7C 示出了示例性片段响应消息 420。片段响应消息 420 包括了 mpd\_content 421 和 fast\_start 推送确认指令 412，包含参数值 K，亦即媒体服务器欲推送片段的个数。

图 7D 示出了媒体客户端 110 与媒体服务器 120 的示例性交互流程。交互流程起始于步骤 431，媒体客户端 110 通过 FDH 访问引擎 113 向所述媒体服务器 120 发送 MPD 请求消息，请求 MPD400，并告诉媒体服务器需要快速启动。在步骤 432 中，媒体服务器 120 接收到 MPD 请求消息后，通过 FDH 服务模块 121 获取预测片段的数目以及具体内容，向媒体客户端 110 发送 MPD 响应消息，所述 MPD 响应消息包含 MPD400 与 push-fast-start acknowledge422。在步骤 433 中，媒体服务器 120 服务模块 121 根据本地预测和 MPD 信息生成预测片段信息，并生成要推送的片段序列。同时向媒体客户端并行发送要推送片段的信息（URL 等信息，不包含具体数据）。在步骤 434 中，所述媒体客户端 110 通过 FDH-DASH 控制引擎 111 解析，所需片段是否在服务器推送的片段消息中，如果在等待获取文件内容，否则，立马请求所需片段。媒体服务器在步骤 435 中，通过服务模块 121 向所述媒体客户端 110 并行发送片段请求消息。在步骤 436 中，所述媒体客户端 110 正常发送片段请求。步骤 434 和步骤 435 可以同时进行，以便于媒体客户端尽早决策。

可选地，根据本申请上述实施例，在所述快速启动类型推送确认指令的推送参数为 URL 模板的情况下，根据将要推送的初始化片段序列和/或媒体片段序列，生成快速启动类型推送确认指令，包括：

所述媒体服务器结合预定数量的媒体客户端和/或所述客户端的历史访问结果确定初始化片段序列和/或媒体片段序列。

所述媒体服务器将所述初始化片段序列和/或媒体片段序列加载至所述快速启动类型推送确认指令；

其中，所述媒体服务器将所述初始化片段序列和/或媒体片段序列推送至所述客户端包括：所述媒体服务器将所述快速启动类型推送确认指令推送至所述客户端。

在一种可选的实施例中，在所述推送确认指令中携带所述媒体服务器预测的推送片段列表的情况下，所述推送片段列表为从所述推送确认指令这种解析得到的所述预测的片段消息，其中，所述媒体服务器根据所述请求消息生成响应信息，包括，

所述媒体服务器结合预定数量的媒体客户端和/或所述客户端的历史访问结果确定预测的推送片段列表；

所述媒体服务器将所述推送片段列表加载至所述推送确认指令之后，得到所述响应信息；

其中，所述媒体服务器将所述预测的片段消息和所述响应信息推送至所述客户端包括：所述媒体服务器将所述推送片段列表推送至所述客户端。

可选地，根据本申请上述实施例，在所述快速启动类型推送确认指令的推送参数为 URL 列表的情况下，根据将要推送的初始化片段序列和/或媒体片段序列，生成快速启动类型推送确认指令，包括：

所述媒体服务器结合预定数量的媒体客户端和/或所述客户端的历史访问结果确定初始化片段序列和/或媒体片段序列。

所述媒体服务器将所述初始化片段序列和/或媒体片段序列加载至所述快速启动类型推送确认指令；

其中，所述媒体服务器将所述初始化片段序列和/或媒体片段序列推送至所述客户端包括：所述媒体服务器将快速启动类型推送确认指令推送至所述客户端。

可选地，根据本申请上述实施例，在所述媒体服务器将所述初始化片段和媒体片段序列的信息推送至所述客户端之后，所述方法还包括：

所述媒体服务器将继续将所述初始化片段和媒体片段序列的信息对应的片段序列推送至所述客户端，其中，所述片段序列为所述客户端需要的媒体内容。

可选地，根据本申请上述实施例，在所述 MPD 请求消息还包括：至少一个请求参数的情况下，所述初始化片段和媒体片段序列的信息包括请求参数所请求的片段的属性。

可选地，根据本申请上述实施例，所述媒体服务器通过如下任意一种或多种方式推送初始化片段和媒体片段序列的信息：HTTP/2 中的推送告知方式和 WebSocket 中的推送内容告知方式。

在一种可选的实施例中，所述推送片段列表为所述媒体服务器预测的片段序列的 URL 组成。

在一种可选的实施例中，在所述推送确认指令中携带所述媒体服务器预测的推送片段信息的情况下，所述推送片段信息为从所述推送确认指令这种解析得到的所述预测的片段消息，其中，所述媒体服务器根据所述请求消息生成响应信息，包括：

所述媒体服务器结合预定数量的媒体客户端和/或所述客户端的历史访问结果确定预测的推送片段信息；

所述媒体服务器将所述推送片段信息加载至所述推送确认指令之后，得到所述响应信息；

其中，所述媒体服务器将所述预测的片段消息和所述响应信息推送至所述客户端包括：所述媒体服务器将所述推送片段信息推送至所述客户端。

在一种可选的实施例中，所述推送片段列消息为所述媒体服务器预测的一个或多个带参数化遍历的 URL 模板和压缩后的参数列表组成。

在一种可选的实施例中，在所述媒体服务器将所述预测的片段消息和所述响应信息推送至所述客户端之后，所述方法还包括，

所述媒体服务器将继续将所述片段消息对应的片段序列推送至所述客户端，其中，所述片段序列为所述客户端需要的媒体内容。

在一种可选的实施例中，在所述请求信息还包括：至少一个请求参数的情况下，所述预测的片段消息包括请求参数所请求的片段的属性。

在一种可选的实施例中，所述媒体服务器通过如下任意一种或多种方式推送所述预测的片段消息至所述媒体服务器：HTTP/2 中的推送告知方式和 WebSocket 中的推送内容告知方式。

本公开实施例中提供了一种流媒体快速启动方法，图 15 是根据本公开实施例的一种流媒体快速启动方法的流程图，图 15 是根据本公开实施例的一种可以由网络服务器(如，媒体服务器 120)实现的流媒体快速启动方法 600 的流程示意图，如图 15 所示，该流程包括如下步骤：

步骤 601，从所述媒体客户端接收 MPD 片段请求消息，其中，所述片段请求消息包含 MPD 请求信息和 push-fast-start directive；

步骤 602，根据服务器端的预测，生成 push-fast-start acknowledge 参数 segment\_list，向所述媒体客户端发送 MPD 片段响应消息，其中，所述片段响应消息包含 push-fast-start acknowledge 和 MPD 片段；

步骤 603，向所述媒体客户端推送上述预测的片段序列。

所述步骤 602 中，所述 push-fast-start acknowledge 参数 URL\_LIST 可以根据服务器所有客户端预测结果来进行预测（亦可以通过初始化指定且不限于所介绍的片段序列指定方法）。

基于以上步骤，媒体服务器只需根据所述 push-fast-start directive，在媒体服务器指定推送片段序列，仅需通过 push-fast-start acknowledge 中的参数 URL\_LIST 列表，即可让客户端快速决策，减少了服务器驱动的盲目性，客户端驱动的不合理性，实现了 FDH-DASH 对媒体客户端的透明性，很好的达到了快速启动的目的。所述方法 600 为示例性实施例，可以根据需要进行修改。

本公开实施例中提供了一种流媒体快速启动方法，图 12 是根据本公开实施例的一种流媒体快速启动方法的流程图，图 12 是根据本公开实施例的一种可以由网络服务器(如，媒体服务器 120)实现的流媒体快速启动方法 900 的流程示意图，如图 12 所示，该流程包括如下步骤：

步骤 901, 从所述媒体客户端接收 MPD 片段请求消息, 其中, 所述片段请求消息包含 MPD 请求信息和 push-fast-start directive;

步骤 902, 根据服务器端的预测 (或者通过结合媒体客户端推送的 push-fast-start directive 参数, 亦即客户端偏好来决策), 生成 push-fast-start acknowledge 参数 URL\_TEMPLATE, 向所述媒体客户端发送 MPD 片段响应消息, 其中, 所述片段响应消息包含 push-fast-start acknowledge 和 MPD 片段;

步骤 903, 向所述媒体客户端推送上述预测的片段序列。

所述步骤 902 中, 所述 push-fast-start acknowledge 参数 URL\_TEMPLATE 可以根据服务器所有客户端预测结果来进行预测 (亦可以通过初始化指定且不限于所介绍的片段序列指定方法)。

基于以上步骤, 媒体服务器只需根据所述 push-fast-start directive, 在媒体服务器指定推送片段序列, 仅需通过 push-fast-start acknowledge 中的参数 URL\_TEMPLATE 列表, 即可让客户快速决策, 减少了服务器驱动的盲目性, 客户端驱动的不合理性, 实现了 FDH-DASH 对媒体客户端的透明性, 很好的达到了快速启动的目的, 同时对于 push-fast-start acknowledge 进行了更加合理的压缩传输, 使得传输参数更加符合规范。所述方法 900 为示例性实施例, 可以根据需要进行修改。

下文是一个示例性实施例, 用以说明在此介绍的流媒体快速启动方法和参数 URL\_TEMPLATE, 及其在一些用例中的潜在应用。

图 9A 示出了示例性 MPD 1000。该 MPD 取自 2016 ICME Grand Challenge 竞赛组委会提供的示例文件 (MPD 文件亦可以是其他标准类型的 DASH MPD 文件)。

图 9B 示出了示例性片段请求消息 1010。片段请求消息 1010 包括了 mpd\_uri 1011 和 fast\_start 推送指令 1012, 不含参数。

图 9B-1 示出了示例性片段请求消息 1010-1。片段请求消息 1010-1 包括了 mpd\_uri 1011-1 和 fast\_start 推送指令 1012-1, 该指令由快速启动信息 (fast-start) 和推送参数组成。所述推送参数有多种, 可以为 video = V、audio = A、init-only、biterate=B 或者它们相互组合 (video、audio、biterate 数值或者范围亦可不指定)。

图 9C 示出了示例性片段响应消息 1020。片段响应消息 1020 包括了 mpd\_content 1021 和 fast\_start 推送确认指令 1012, 包含参数 URL\_TEMPLATE, 亦即媒体服务器欲推送片段的信息列表压缩形式 (此处可以为 FDH-DASH 标准中的定义格式等)。

图 9D 示出了媒体客户端 110 与媒体服务器 120 的示例性交互流程。交互流程起始于步骤 1031, 媒体客户端 110 通过 FDH 访问引擎 113 向所述媒体服务器 120 发送 MPD 请求消息, 请求 MPD1000, 并告诉媒体服务器需要快速启动。在步骤 1032 中, 媒体服务器 120 接收到 MPD 请求消息后, 通过 FDH 服务模块 121 获取预测片段的数目以及具体内容, 向媒体客户端

110 发送 MPD 响应消息,所述 MPD 响应消息包含 MPD400 与 push-fast-start acknowledge1022。在步骤 1033 中,所述媒体客户端 110 通过 FDH-DASH 控制引擎 111 解析,所需片段是否在服务器推送的片段列表中,如果在等待获取文件内容,否则,立马请求所需片段。媒体服务器在步骤 1034 中,通过服务模块 121 向所述媒体客户端 110 并行发送片段请求消息。在步骤 1035 中,所述媒体客户端 110 正常发送片段请求。步骤 1033 和步骤 1034 可以同时进行,以便于媒体客户端尽早决策。

图 13 是计算机系统或网络设备 1200 的实施例的示意图。所述网络设备 1200 可以为任意合适的设备予以实现,如,网络服务器、媒体服务器(如,媒体服务器 120)或用户端设备(如,媒体客户端 110)等等。所述网络设备 1200 能够接收、处理和发送去向以及来自网络的片段请求消息/片段响应消息/片段等消息。所述网络设备 1200 可以包括与接收器 1211 连接的一个或多个入口端 1210,其中,所述接收器 1211 可设置为通过 FDH 通道从其他网络节点接收 MPD 和/或片段。所述网络设备 1200 还可以包括与发送器 1231 连接的一个或多个出口端 1230,其中,所述发送器 1231 可以通过 FDH 通道向其他网络节点发送 MPD 和/或片段。所述网络设备 1200 还可以包括处理器 1220,其中,所述处理器 1220 与所述接收器 1211 链接,设置为处理所述片段或者确定向那个网络节点发送所述片段。

所述处理器 1220 可以通过硬件、软件及其组合来实现。所述处理器 1220 可用一个或多个 CPU 芯片、可编程门阵列、专用集成电路和/或数字信号处理器予以实现。所述处理 1220 可用于实现在此描述的任意功能模块或单元,例如,FDH-DASH 控制引擎 111、FDH 访问引擎 113、媒体引擎 112、FDH 服务模块 121、解析引擎 122、媒体播放器 1221 或者本领域普通技术人员熟知的其他任何功能组件,或者其任何组合。所述媒体播放器 1221 可以为在所述处理器 1220 上运行的应用程序,设置为解码播放所述媒体片段。

所述网络设备 1200 还可以包括至少一个存储器 1222。所述存储器 1222 可设置为存储 MPD1223 和片段 1224。所述 MPD1223 和所述片段 1224 可以通过以上所述的各种形式进行存储和打包。例如,所述 MPD1223 可以为所述 MPD300(或者其他在 MPEG-DASH 中定义的标准 MPD 文件)。如图 15 所示,所述片段 1224 和所述 MPD1223 可以作为单独的文件进行存储,也可以将其存储在单个文件中。

在实践中,可能存在所述网络设备 1200 处理的双向流量,因此,一些端口可既接收也发送报文。在这种情况下,所述入口端 1210 和出口端 1230 可以是共设在一起的或者可以认识始于收发器连接的相同端口的不同功能。所述处理器 1220、所述存储器 1222、所述接收器 1211 和所述发送器 1231 中的一个或多个也可以设置为至少部分实现或支持上文描述的任意方法和实现方式。

通过以上的实施方式的描述,本领域的技术人员可以清楚地了解到根据上述实施例的方法可借助软件加必需的通用硬件平台的方式来实现,当然也可以通过硬件,但很多情况下前者是更佳的实施方式。基于这样的理解,本公开的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质(如 ROM/RAM、磁碟、光盘)中,包括若干指令用以使得一台终端设备(可以是手机,计算机,

服务器，或者网络设备等）执行本公开各个实施例所述的方法。

在本实施例中还提供了一种流媒体快速启动装置，该装置用于实现上述实施例及优选实施方式，已经进行过说明的不再赘述。如以下所使用的，术语“模块”可以实现预定功能的软件和/或硬件的组合。尽管以下实施例所描述的装置较佳地以软件来实现，但是硬件，或者软件和硬件的组合的实现也是可能并被构想的。

图 17 是根据本公开实施例的流媒体快速启动装置的结构框图，如图 17 所示，该装置包括：

访问引擎模块 170，设置为客户端向媒体服务器发送快速启动类型推送指令，其中，所述快速启动类型推送指令用于指示所述媒体服务器推送初始化片段序列和/或媒体片段序列。

第一接收模块 172，设置为所述客户端接收所述媒体服务器返回的快速启动类型推送确认指令。

控制引擎模块 174，设置为所述客户端根据所述快速启动类型推送确认指令获取所述媒体服务器将要推送的初始化片段和/或媒体片段序列的信息。

在一种可选的实施例中，上述流媒体快速启动装置包括：生成引擎模块，设置为客户端生成快速启动类型推送指令 `push-fast-start directive`；访问引擎模块，设置为向媒体服务器发送所述快速启动类型推送指令，其中，所述快速启动类型推送指令用于通知所述媒体服务器推送初始化片段序列和/或媒体片段序列；第一接收模块，设置为所述客户端获取快速启动类型推送确认指令 `push-fast-start acknowledge`。控制引擎模块，设置为所述客户端根据所述快速启动类型推送确认指令获取所述服务器将要推送的初始化片段和媒体片段序列的信息。

在另一种可选的实施例中，上述流媒体快速启动装置包括：生成引擎模块，设置为客户端生成快速启动推动指令；访问引擎模块，设置为客户端发送用于获取媒体呈现描述文件 MPD 的请求消息至媒体服务器，其中，所述请求信息至少包括：快速启动推送指令和 MPD 请求指令；接收模块，设置为所述客户端接收所述媒体服务器根据所述请求消息返回的响应消息，其中，所述响应消息包括：确定所述客户端需要快速启动的推送确认指令、响应所述 MPD 请求指令得到的 MPD 数据；控制引擎模块，设置为在所述客户端获取到预测的片段消息之后，根据所述预测的片段消息和所述 MPD 数据，获取所述客户端需要的媒体内容，其中，所述预测的片段消息为所述媒体服务器主动推送的与所述推送确认指令对应的片段消息。

其中，上述生成引擎模块可以是方法实施例中的生成引擎，访问引擎模块可以是方法实施例中的访问引擎模块，控制引擎模块可以是方法实施例中的控制引擎。

可选地，根据本申请上述实施例，上述第一接收模块还包括：

第一接收子模块，设置为所述客户端接收所述媒体服务器返回的 MPD 响应消息，所述 MPD 响应消息至少包含 MPD 数据，以及所述快速启动类型推送确认指令。

可选地，根据本申请上述实施例，所述控制引擎模块包括：

第一确认模块，设置为所述快速启动类型推送确认指令中携带 URL 模板 URL\_TEMPLATE 类型的推送参数；

第一获取模块，设置为所述客户端解析 URL 模板推送参数，获取所述媒体服务器将要推送的初始化片段序列和/或媒体片段序列的信息。

可选地，根据本申请上述实施例，所述控制引擎模块包括：

第二确认模块，设置为所述快速启动类型推送确认指令中携带数值 NUMBER 类型的推送参数；

第二获取模块，设置为所述客户端解析数值推送参数 K，并接收后续的 K 个推送承诺 PUSH\_PROMISE 帧，获取所述媒体服务器将要推送的初始化片段序列和/或媒体片段序列的信息。

可选地，根据本申请上述实施例，所述控制引擎模块包括：

第三确认模块，设置为所述快速启动类型推送确认指令中携带 URL 列表 URL\_LIST 类型的推送参数；

第三获取模块，设置为所述客户端解析 URL 列表推送参数，获取所述媒体服务器将要推送的初始化片段序列和/或媒体片段序列的信息。

可选地，根据本申请上述实施例，所述装置还包括：

第四确认模块，设置为所述客户端解析所述 MPD 数据，确定需要所述媒体服务器推送的初始化片段和/或媒体片段序列的信息；

第二接收模块，设置为若所述获取的媒体服务器将要推送的初始化片段和/或媒体片段序列的信息等同于所述确定的需要媒体服务器推送的初始化片段和/或媒体片段序列的信息，所述客户端接收所述媒体服务器推送的初始化片段序列和/或媒体片段序列。

在一种可选的实施例中，该装置除包括图 17 所示的所有模块外，控制引擎模块包括：

发送模块，设置为通过 MPD 请求消息向所述媒体服务器发送所述快速启动类型推送指令，其中，所述 MPD 请求消息用于请求所述媒体服务器发送 MPD 响应消息，所述 MPD 响应消息至少应包含 MPD 数据和快速启动类型推送确认指令。

在一种可选的实施例中，上述引擎模块包括：确认模块，设置为在所述推送确认指令中携带所述媒体服务器预测的参数值 K 的情况下，所述客户端从所述推送确认指令中解析得到所述参数值 K，并接收所述媒体服务器推送的与所述参数值 K 对应的 K 个片段消息。

在一种可选的实施例中，上述第一接收模块还包括：第一获取模块，设置为通过获取并解析 MPD 响应消息获取所述快速启动类型推送确认指令，所述 MPD 响应消息至少包含 MPD 数据和快速启动类型推送确认指令。

在一种可选的实施例中，根据本申请上述实施例，上述控制引擎模块包括：

初始化模块，设置为初始化片段序列和/或媒体片段序列的信息；

第二获取模块，设置为在所述快速启动类型推送指令的推送参数 **PUSH\_PARAMS** 为数值 **NUMBER** 类型的情况下，所述客户端通过解析数值 **K** 和后续的 **K** 个推送承诺 **PUSH\_PROMISE** 帧，获取媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息；

第三获取模块，设置为在所述快速启动类型推送指令的推送参数 **PUSH\_PARAMS** 为 **URL** 列表 **URL\_LIST** 类型的情况下，所述客户端解析 **URL** 列表，获取媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息。

第一获取模块，设置为通过获取并解析 **MPD** 响应消息获取所述快速启动类型推送确认指令，所述 **MPD** 响应消息至少包含 **MPD** 数据和快速启动类型推送确认指令

在一种可选的实施例中，该装置除包括图 17 所示的所有模块外，控制引擎模块还包括：

第一解析引擎模块，设置为所述客户端解析所述 **MPD** 数据，得到所述客户端需要的媒体内容的媒体信息；

第一检测模块，设置为所述客户端检测所述 **K** 个片段消息中是否命中所述媒体信息，所述片段消息为 **URL** 地址；

第一获取模块，设置为如果在所述 **K** 个片段消息中命中所述媒体消息成功，所述客户端获取所述媒体服务器推送的与所述预测的片段消息所对应的片段序列，其中，所述片段序列为所述客户端需要的媒体内容。

在一种可选的实施例中，根据本申请上述实施例，上述第二获取模块包括：

第二接收模块，设置为所述客户端从所述快速启动类型推送指令中解析得到所述参数值 **K**，并接收所述媒体服务器推送的与所述参数值 **K** 对应的 **K** 个片段消息；

第二解析引擎模块，设置为所述客户端解析所述 **MPD** 数据，得到所述客户端需要的媒体内容的媒体信息；

第二检测模块，设置为所述客户端检测所述 **K** 个片段消息中是否命中所述媒体信息，所述片段消息为 **URL** 地址；

第四获取模块，设置为如果在所述 **K** 个片段消息中命中所述媒体消息成功，所述客户端获取所述媒体服务器将要推送的初始化片段和媒体片段序列的信息。

在一种可选的实施例中，如果在所述 **K** 个片段消息中命中所述媒体消息失败，所述客户端发起获取所述需要的媒体内容的请求，或取消所述媒体服务器推送媒体内容的会话。

在一种可选的实施例中，在所述推送确认指令中携带所述媒体服务器预测的推送片段列表的情况下，所述推送片段列表为从所述推送确认指令这种解析得到的所述预测的片段消息，

其中，控制引擎模块包括：

第二解析引擎模块，所述客户端解析所述 MPD 数据，得到所述客户端需要的媒体内容的媒体信息；

第二检测模块，设置为所述客户端检测所述推送片段列表中是否命中所述媒体信息；

第二获取模块，设置为如果在所述推送片段列表中命中所述媒体消息成功，所述客户端获取所述媒体服务器推送的与所述推送片段列表所对应的片段序列，其中，所述片段序列为所述客户端需要的媒体内容。

图 18 是根据本公开实施例的流媒体快速启动装置的结构框图，如图 18 所示，该装置包括：

服务模块 180，设置为媒体服务器接收客户端发送的快速启动类型推送指令。

生成引擎模块 182，设置为所述媒体服务器根据将要推送的初始化片段和/或媒体片段序列，生成快速启动类型推送确认指令。

第一推送模块 184，设置为所述媒体服务器向所述客户端发送所述快速启动类型推送确认指令。

在一种可选的实施例中，该装置包括：服务模块，设置为媒体服务器获取快速启动类型推送指令 `push-fast-start directive`；生成引擎模块，设置为所述媒体服务器生成快速启动类型推送确认指令 `push-fast-start acknowledge`，其中，所述快速启动类型推送确认指令用于通知客户端将要推送的推送初始化片段序列和/或媒体片段序列；第一推送模块，设置为向所述客户端发送所述快速启动类型推送确认指令。

在一种可选的实施例中，上述流媒体快速启动装置可以包括：服务模块，设置为媒体服务器接收客户端发送的用于获取媒体呈现描述文件 MPD 的请求消息，其中，所述请求信息至少包括：快速启动推送指令和 MPD 请求指令；生成引擎模块，所述媒体服务器根据所述请求消息生成响应信息，其中，所述响应消息包括：确定所述客户端需要快速启动的推送确认指令、响应所述 MPD 请求指令得到的 MPD 数据；解析引擎模块，设置为在所述媒体服务器生成预测的片段消息之后，所述媒体服务器将所述预测的片段消息和所述响应信息推送至所述客户端，使得所述客户端根据所述预测的片段消息和所述 MPD 数据，获取所述客户端需要的媒体内容，其中，所述预测的片段消息为所述媒体服务器主动推送的与所述推送确认指令对应的片段消息。

其中，上述服务模块可以是方法实施例中的服务模块，上述生成引擎模块可以是方法实施例中的生成引擎，上述解析引擎模块可以是方法实施例中的解析引擎。

在本实施例中还提供了一种流媒体快速启动系统，该系统设置为实现上述实施例及优选实施方式，已经进行过说明的不再赘述。

图 19 是根据本公开实施例的流媒体快速启动系统的结构框图，如图 19 所示，该系统包括：

客户端 190，设置为向媒体服务器发送所述快速启动类型推送指令，并接收媒体服务器返回的快速启动类型推送确认指令，根据所述快速启动类型推送确认指令获取所述服务器将要推送的初始化片段和/或媒体片段序列的信息，其中，所述快速启动类型推送指令用于指示所述媒体服务器推送初始化片段序列和/或媒体片段序列；

所述媒体服务器 192，与所述客户端通信，设置为接收所述客户端发送的所述快速启动类型推送指令，生成快速启动类型推送确认指令，向所述客户端发送所述快速启动类型推送确认指令，并向所述客户端推送所述初始化片段序列和/或所述媒体片段序列。

在一种可选的实施例中，上述系统包括：客户端，设置为生成快速启动类型推送指令 `push-fast-start directive`，并向媒体服务器发送所述快速启动类型推送指令，在获取到快速启动类型推送确认指令 `push-fast-start acknowledge` 之后，根据所述快速启动类型推送确认指令获取所述服务器将要推送的初始化片段和媒体片段序列的信息，其中，所述快速启动类型推送指令用于通知所述媒体服务器推送初始化片段序列和/或媒体片段序列；所述媒体服务器，与所述客户端通信，设置为接收所述客户端发送的所述快速启动类型推送指令，向所述客户端发送所述快速启动类型推送确认指令，并向所述客户端推送所述初始化片段序列和/或所述媒体片段序列。

在另一种可选的实施例中，上述流媒体快速启动系统包括：客户端，生成快速启动推动指令，并向媒体服务器发送用于获取媒体呈现描述文件 MPD 的请求消息至媒体服务器，在获取到预测的片段消息之后，根据所述预测的片段消息和所述 MPD 数据，获取需要的媒体内容，其中，所述请求信息至少包括：快速启动推送指令和 MPD 请求指令，所述预测的片段消息为所述媒体服务器主动推送的与所述推送确认指令对应的片段消息。媒体服务器，与所述客户端通信，设置为接收客户端发送的用于获取媒体呈现描述文件 MPD 的请求消息，并根据所述请求消息生成响应信息，在生成预测的片段消息之后，将所述预测的片段消息和所述响应信息推送至所述客户端。

需要说明的是，上述各个模块是可以通过软件或硬件来实现的，对于后者，可以通过以下方式实现，但不限于此：上述模块均位于同一处理器中；或者，上述模块分别位于多个处理器中。

本公开的实施例还提供了一种存储介质，该存储介质包括存储的程序，其中，上述程序运行时执行上述任一项的方法。

可选地，在本实施例中，上述存储介质可以被设置为存储用于执行以下步骤的程序代码：

S1，客户端向媒体服务器发送快速启动类型推送指令，其中，所述快速启动类型推送指令用于指示所述媒体服务器推送初始化片段序列和/或媒体片段序列。

S2，所述客户端接收所述媒体服务器返回的快速启动类型推送确认指令。

S3, 所述客户端根据所述快速启动类型推送确认指令获取所述媒体服务器将要推送的初始化片段和/或媒体片段序列的信息。

可选地, 存储介质还被设置为存储用于执行以下步骤的程序代码: 客户端向媒体服务器发送快速启动类型推送指令包括: 所述客户端通过 MPD 请求消息向所述媒体服务器发送所述快速启动类型推送指令。

可选地, 存储介质还被设置为存储用于执行以下步骤的程序代码: 所述客户端接收所述快速启动类型推送确认指令包括: 所述客户端接收所述媒体服务器返回的 MPD 响应消息, 所述 MPD 响应消息至少包含 MPD 数据, 以及所述快速启动类型推送确认指令。

可选地, 存储介质还被设置为存储用于执行以下步骤的程序代码: 所述客户端根据所述快速启动类型推送确认指令获取所述服务器将要推送的初始化片段和/或媒体片段序列的信息, 包括: 所述快速启动类型推送确认指令中携带 URL 模板类型的推送参数; 所述客户端解析 URL 模板推送参数, 获取所述媒体服务器将要推送的初始化片段序列和/或媒体片段序列的信息。

可选地, 存储介质还被设置为存储用于执行以下步骤的程序代码: 所述客户端根据所述快速启动类型推送确认指令获取所述服务器将要推送的初始化片段和/或媒体片段序列的信息, 包括: 所述快速启动类型推送确认指令中携带数值类型的推送参数; 所述客户端解析数值推送参数 K, 并接收后续的 K 个推送承诺 PUSH\_PROMISE 帧, 获取所述媒体服务器将要推送的初始化片段序列和/或媒体片段序列的信息。

可选地, 存储介质还被设置为存储用于执行以下步骤的程序代码: 所述客户端根据所述快速启动类型推送确认指令获取所述服务器将要推送的初始化片段和/或媒体片段序列的信息, 包括: 所述快速启动类型推送确认指令中携带 URL 列表类型的推送参数; 所述客户端解析 URL 列表推送参数, 获取所述媒体服务器将要推送的初始化片段序列和/或媒体片段序列的信息。

可选地, 存储介质还被设置为存储用于执行以下步骤的程序代码: 所述客户端解析所述 MPD 数据, 确定需要所述媒体服务器推送的初始化片段和/或媒体片段序列的信息; 若所述获取的媒体服务器将要推送的初始化片段和/或媒体片段序列的信息等同于所述确定的需要媒体服务器推送的初始化片段和/或媒体片段序列的信息, 所述客户端接收所述媒体服务器推送的初始化片段序列和/或媒体片段序列。

可选地, 存储介质还被设置为存储用于执行以下步骤的程序代码: 所述客户端解析所述 MPD 数据, 确定需要所述媒体服务器推送的初始化片段和/或媒体片段序列的信息; 若所述获取的媒体服务器将要推送的初始化片段和/或媒体片段序列的信息不同于所述确定的需要媒体服务器推送的初始化片段和/或媒体片段序列的信息, 所述客户端取消所述媒体服务器推送的初始化片段序列和/或媒体片段序列。

可选地, 存储介质还被设置为存储用于执行以下步骤的程序代码: 所述客户端向所述媒

体服务器发送获取所述确定需要所述媒体服务器推送的初始化片段和/或媒体片段序列的请求。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：所述 MPD 请求消息中携带的快速启动推送指令包含偏好参数信息，所述偏好参数包括如下任意一种或多种属性：分辨率范围、码率范围、媒体属性和语言。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：所述客户端接收所述媒体服务器返回的 MPD 响应消息，以及所述快速启动类型推送确认指令，进一步包括：通过 HTTP/2 报文方式接收所述 MPD 响应消息，其中，所述 HTTP/2 报文的 Accept-Push-Policy 首部携带快速启动类型推送确认指令；或者，通过 WebSocket 中 DASH 子协议的新新\_mpd 消息接收所述 MPD 响应消息，其中，所述 new\_mpd 消息的 JSON 键值对携带快速启动类型推送确认指令。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：所述客户端向所述媒体服务器发送 MPD 请求消息，以及所述快速启动类型推送指令，进一步包括：通过 HTTP/2 报文方式发送所述 MPD 请求消息，其中，所述 HTTP/2 报文的 Accept-Push-Policy 首部携带快速启动类型推送指令；或者，通过 WebSocket 中 DASH 子协议的 get\_mpd 消息发送所述 MPD 请求消息，其中，所述 get\_mpd 消息的 JSON 键值对携带快速启动类型推送确认指令。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：所述客户端通过如下任意一种或多种方式从所述媒体服务器获取到所述 MPD 响应消息：HTTP/2 的 MPD 响应报文方式和 WebSocket 中 DASH 子协议的新新\_mpd 消息，其中，HTTP/2 的 MPD 响应报文通过 Push-Policy 首部携带快速启动类型推送确认指令；WebSockets 中 DASH 子协议的新新\_mpd 消息通过 JSON 键值对形式携带快速启动类型推送确认指令。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：所述客户端通过如下任意一种或多种方式发送所述 MPD 请求消息：HTTP/2 请求报文方式和 WebSocket 中 DASH 子协议的 get\_mpd 消息，其中，HTTP/2 请求报文通过 Accept-Push-Policy 首部携带快速启动类型推送指令；WebSockets 中 DASH 子协议的 get\_mpd 消息通过 JSON 键值对形式携带快速启动类型推送指令。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：媒体服务器接收客户端发送的快速启动类型推送指令，其中，所述快速启动类型推送指令用于指示所述媒体服务器推送初始化片段序列和/或媒体片段序列；所述媒体服务器根据将要推送的初始化片段和/或媒体片段序列，生成快速启动类型推送确认指令；所述媒体服务器向所述客户端发送所述快速启动类型推送确认指令。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：媒体服务器接收快速启动类型推送指令包括：所述媒体服务器接收所述客户端发送的 MPD 请求消息，所述 MPD 请求消息中包含所述快速启动类型推送指令。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：所述媒体服务器生成快速启动类型推送确认指令，包括：所述媒体服务器根据将要推送的初始化片段序列和/或媒体片段序列，生成 URL 模板类型的推送参数；所述快速启动类型推送确认指令中携带所述 URL 模板类型的推送参数。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：所述媒体服务器生成快速启动类型推送确认指令，包括：所述媒体服务器根据将要推送的初始化片段序列和/或媒体片段序列，生成数值类型的推送参数；所述快速启动类型推送确认指令中携带所述数值类型的推送参数。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：所述媒体服务器生成快速启动类型推送确认指令，包括：所述媒体服务器根据将要推送的初始化片段序列和/或媒体片段序列，生成 URL 列表类型的推送参数；所述快速启动类型推送确认指令中携带所述 URL 列表类型的推送参数。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：发送所述快速启动类型推送确认指令包括：所述媒体服务器通过 MPD 响应消息向所述客户端发送所述快速启动类型推送确认指令，其中，所述 MPD 响应消息至少包含 MPD 数据，以及所述快速启动类型推送确认指令。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：所述媒体服务器生成快速启动类型推送确认指令，包括：所述媒体服务器根据所述客户端的预定数量和/或所述客户端的历史访问结果，确定所述快速启动类型推送确认指令中携带的推送参数。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：所述媒体服务器向所述客户端发送所述快速启动类型推送确认指令之后，所述方法还包括：所述媒体服务器向所述客户端推送所述快速启动类型推送确认指令指示的片段序列初始化片段和/或媒体片段序列。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：所述 MPD 请求消息中携带的快速启动推送指令包含偏好参数信息，所述偏好参数包括如下任意一种或多种属性：分辨率范围、码率范围、媒体属性和语言。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：所述媒体服务器通过如下任意一种或多种方式推送初始化片段和媒体片段序列的信息：HTTP/2 的 MPD 响应报文方式和 WebSocket 中 DASH 子协议的新新\_mpd 消息，其中，HTTP/2 的 MPD 响应报文通过 Push-Policy 首部携带快速启动类型推送确认指令；WebSockets 中 DASH 子协议的新新\_mpd 消息通过 JSON 键值对形式携带快速启动类型推送确认指令。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：所述媒体服务器接收所述客户端发送的 MPD 请求消息，以及所述快速启动类型推送指令，进一步包括：通过 HTTP/2 报文方式接收所述 MPD 请求消息，其中，所述 HTTP/2 报文的 Accept-Push-Policy 首部携带快

速启动类型推送指令；或者，通过 WebSocket 中 DASH 子协议的 `get_mpd` 消息接收所述 MPD 请求消息，其中，所述 `get_mpd` 消息的 JSON 键值对携带快速启动类型推送确认指令。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：所述媒体服务器向所述客户端发送 MPD 响应消息，以及所述快速启动类型推送确认指令，进一步包括：通过 HTTP/2 报文方式发送所述 MPD 响应消息，其中，所述 HTTP/2 报文的 `Accept-Push-Policy` 首部携带快速启动类型推送确认指令；或者，通过 WebSocket 中 DASH 子协议的 `new_mpd` 消息发送所述 MPD 响应消息，其中，所述 `new_mpd` 消息的 JSON 键值对携带快速启动类型推送确认指令。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：客户端生成快速启动类型推送指令 `push-fast-start directive`；向媒体服务器发送所述快速启动类型推送指令，其中，所述快速启动类型推送指令用于通知所述媒体服务器推送初始化片段序列和/或媒体片段序列；所述客户端获取快速启动类型推送确认指令 `push-fast-start acknowledge`；所述客户端根据所述快速启动类型推送确认指令获取所述服务器将要推送的初始化片段和媒体片段序列的信息。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：向媒体服务器发送快速启动类型推送指令包括：通过 MPD 请求消息向所述媒体服务器发送所述快速启动类型推送指令，其中，所述 MPD 请求消息用于请求所述媒体服务器发送 MPD 响应消息，所述 MPD 响应消息至少应包含 MPD 数据和快速启动类型推送确认指令。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：所述客户端获取所述快速启动类型推送确认指令包括：通过获取并解析 MPD 响应消息获取所述快速启动类型推送确认指令，所述 MPD 响应消息至少包含 MPD 数据和快速启动类型推送确认指令。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：所述客户端根据所述快速启动类型推送确认指令获取所述服务器将要推送的初始化片段和媒体片段序列的信息：在所述快速启动类型推送指令的推送参数 `PUSH_PARAMS` 为 URL 模板 `URL_TEMPLATE` 类型的情况下，所述客户端解析 URL 模板，获取媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息；在所述快速启动类型推送指令的推送参数 `PUSH_PARAMS` 为数值 `NUMBER` 类型的情况下，所述客户端通过解析数值 `K` 和后续的 `K` 个推送承诺 `PUSH_PROMISE` 帧，获取媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息；在所述快速启动类型推送指令的推送参数 `PUSH_PARAMS` 为 URL 列表 `URL_LIST` 类型的情况下，所述客户端解析 URL 列表，获取媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：在所述快速启动类型推送指令的推送参数 `PUSH_PARAMS` 为数值 `NUMBER` 类型的情况下，所述客户端通过解析数值 `K` 和后续的 `K` 个推送承诺 `PUSH_PROMISE` 帧，获取媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息，包括：所述客户端从所述快速启动类型推送指令中解析得到所述参数值 `K`，并接收所述媒体服务器推送的与所述参数值 `K` 对应的 `K` 个片段消息；所述客户端解析所述 MPD 数据，得到所述客户端需要的媒体内容的媒体信息；所述客户端检测所述

K 个片段消息中是否命中所述媒体信息，所述片段消息为 URL 地址；如果在所述 K 个片段消息中命中所述媒体消息成功，所述客户端获取所述媒体服务器将要推送的初始化片段和媒体片段序列的信息。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：如果在所述 K 个片段消息中命中所述媒体消息失败，所述客户端发起获取所述需要的媒体内容的请求并取消所述媒体服务器推送媒体内容的会话，或所述客户端发起获取所述需要的媒体内容的请求并存储所述媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：在所述快速启动类型推送指令的推送参数 PUSH\_PARAMS 为 URL 模板 URL\_TEMPLATE 类型的情况下，所述客户端解析 URL 模板，获取媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息，包括：所述客户端解析所述 MPD 数据，得到所述客户端需要的媒体内容的媒体信息；所述客户端检测所述媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息是否命中所述媒体信息；如果在所述媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息命中所述媒体消息成功，获取媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：如果在所述媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息命中所述媒体消息失败，所述客户端发起获取所述需要的媒体内容的请求并取消所述媒体服务器推送媒体内容的会话，或所述客户端发起获取所述需要的媒体内容的请求并存储所述媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：在所述快速启动类型推送指令的推送参数 PUSH\_PARAMS 为 URL 列表 URL\_LIST 类型的情况下，所述客户端解析 URL 列表，获取媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息，包括，所述客户端解析所述 MPD 数据，得到所述客户端需要的媒体内容的媒体信息；所述客户端检测所述媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息是否命中所述媒体信息；如果在所述媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息命中所述媒体消息成功，获取媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：如果在所述媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息命中所述媒体消息失败，所述客户端发起获取所述需要的媒体内容的请求并取消所述媒体服务器推送媒体内容的会话，或所述客户端发起获取所述需要的媒体内容的请求并存储所述媒体服务器将要的推送初始化片段序列和/或媒体片段序列的信息。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：在所述 MPD 请求消息还包括：至少一个请求参数的情况下，所述服务器将要推送的初始化片段和媒体片段序列的信息包括所述 MPD 请求消息的片段的属性。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：所述客户端通过如下

任意一种或多种方式从所述媒体服务器获取到所述服务器将要推送的初始化片段和媒体片段序列的信息：HTTP/2 中的推送告知方式和 WebSocket 中的推送内容告知方式。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：所述客户端通过如下任意一种或多种方式发送所述 MPD 请求消息：HTTP/2 请求报文的扩展首部的键值对形式和 WebSocket 的帧中的 JSON 键值对形式。

上述存储介质可以被设置为存储用于执行以下步骤的程序代码：媒体服务器获取快速启动类型推送指令 `push-fast-start directive`；所述媒体服务器生成快速启动类型推送确认指令 `push-fast-start acknowledge`，其中，所述快速启动类型推送确认指令用于通知客户端将要推送的推送初始化片段序列和/或媒体片段序列；向所述客户端发送所述快速启动类型推送确认指令；向所述客户端推送初始化片段序列和/或媒体片段序列。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：媒体服务器获取快速启动类型推送指令 `push-fast-start directive` 包括：通过获取并解析 MPD 请求消息获取所述快速启动类型推送指令，所述 MPD 请求消息至少包含 MPD URI 和快速启动类型推送指令。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：所述媒体服务器生成快速启动类型推送确认指令 `push-fast-start acknowledge`：所述媒体服务器根据将要推送的初始化片段序列和/或媒体片段序列，生成快速启动类型推送确认指令，其中，所述快速启动类型推送确认指令的推送参数可为 URL 模板 `URL_TEMPLATE` 类型，和/或 数值 `NUMBER` 类型，和/或 URL 列表 `URL_LIST` 类型。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：向所述客户端发送所述快速启动类型推送确认指令包括：通过 MPD 响应消息向所述客户端发送所述快速启动类型推送确认指令，其中，所述 MPD 响应消息用于响应所述客户端发送 MPD 请求消息，所述 MPD 响应消息至少包含 MPD 数据和所述快速启动类型推送确认指令。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：在所述快速启动类型推送确认指令的推送参数为数值 `NUMBER` 类型的情况下，根据将要推送的初始化片段序列和/或媒体片段序列，生成快速启动类型推送确认指令，包括：所述媒体服务器结合预定数量的媒体客户端和/或所述客户端的历史访问结果确定数值 `K`，并确定与所述数值 `K` 对应的 `K` 个片段消息；所述媒体服务器将所述数值 `K` 加载至所述快速启动类型推送确认指令；其中，所述媒体服务器将所述初始化片段序列和/或媒体片段序列推送至所述客户端包括：所述媒体服务器将所述数值 `K` 以及与所述数值 `K` 对应的 `K` 个片段消息推送至所述客户端。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：在所述快速启动类型推送确认指令的推送参数为 URL 模板的情况下，根据将要推送的初始化片段序列和/或媒体片段序列，生成快速启动类型推送确认指令，包括：所述媒体服务器结合预定数量的媒体客户端和/或所述客户端的历史访问结果确定初始化片段序列和/或媒体片段序列；所述媒体服务器将所述初始化片段序列和/或媒体片段序列加载至所述快速启动类型推送确认指令；其中，所述媒体服务器将所述初始化片段序列和/或媒体片段序列推送至所述客户端包括：所述媒体服

务器将所述快速启动类型推送确认指令推送至所述客户端。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：在所述快速启动类型推送确认指令的推送参数为 URL 列表的情况下，根据将要推送的初始化片段序列和/或媒体片段序列，生成快速启动类型推送确认指令，包括：所述媒体服务器结合预定数量的媒体客户端和/或所述客户端的历史访问结果确定初始化片段序列和/或媒体片段序列；所述媒体服务器将所述初始化片段序列和/或媒体片段序列加载至所述快速启动类型推送确认指令；其中，所述媒体服务器将所述初始化片段序列和/或媒体片段序列推送至所述客户端包括：所述媒体服务器将快速启动类型推送确认指令推送至所述客户端。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：在所述媒体服务器将所述初始化片段和媒体片段序列的信息推送至所述客户端之后，所述方法还包括：所述媒体服务器将继续将所述初始化片段和媒体片段序列的信息对应的片段序列推送至所述客户端，其中，所述片段序列为所述客户端需要的媒体内容。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：在所述 MPD 请求消息还包括：至少一个请求参数的情况下，所述初始化片段和媒体片段序列的信息包括请求参数所请求的片段的属性。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：所述媒体服务器通过如下任意一种或多种方式推送初始化片段和媒体片段序列的信息：HTTP/2 中的推送告知方式和 WebSocket 中的推送内容告知方式。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：客户端从推送确认指令中解析得到参数值 K，并接收媒体服务器推送的与参数值 K 对应的 K 个片段消息。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：客户端解析 MPD 数据，得到客户端需要的媒体内容的媒体信息；客户端检测 K 个片段消息中是否命中媒体信息，片段消息为 URL 地址；如果在 K 个片段消息中命中媒体消息成功，客户端获取媒体服务器推送的与预测的片段消息所对应的片段序列，其中，片段序列为客户端需要的媒体内容。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：如果在 K 个片段消息中命中媒体消息失败，客户端发起获取需要的媒体内容的请求，或取消媒体服务器推送媒体内容的会话。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：在推送确认指令中携带媒体服务器预测的推送片段列表的情况下，推送片段列表为从推送确认指令这种解析得到的预测的片段消息，其中，根据预测的片段消息和 MPD 数据，获取客户端需要的媒体内容，包括：客户端解析 MPD 数据，得到客户端需要的媒体内容的媒体信息；客户端检测推送片段列表中是否命中媒体信息；如果在推送片段列表中命中媒体消息成功，客户端获取媒体服务器推送的与推送片段列表所对应的片段序列，其中，片段序列为客户端需要的媒体内容。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：如果在推送片段列表

命中媒体消息失败，客户端发起获取需要的媒体内容的请求，或取消媒体服务器推送媒体内容的会话。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：推送片段列表为媒体服务器预测的片段序列的 URL 组成。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：在推送确认指令中携带媒体服务器预测的推送片段信息的情况下，推送片段信息为从推送确认指令这种解析得到的预测的片段消息，其中，根据预测的片段消息和 MPD 数据，获取客户端需要的媒体内容，包括：客户端解析 MPD 数据，得到客户端需要的媒体内容的媒体信息；客户端检测推送片段消息中是否命中媒体信息；如果在推送片段消息中命中媒体消息成功，客户端获取媒体服务器推送的与推送片段消息所对应的片段序列，其中，片段序列为客户端需要的媒体内容。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：如果在推送片段消息中命中媒体消息失败，客户端发起获取需要的媒体内容的请求，或取消媒体服务器推送媒体内容的会话。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：推送片段列消息为媒体服务器预测的一个或多个带参数化遍历的 URL 模板和压缩后的参数列表组成。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：在请求信息还包括：至少一个请求参数的情况下，预测的片段消息包括请求参数所请求的片段的属性。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码：客户端通过如下任意一种或多种方式从媒体服务器获取到预测的片段消息：HTTP/2 中的推送告知方式和 WebSocket 中的推送内容告知方式。

可选地，存储介质还被设置为存储用于执行以下步骤的程序代码客户端通过如下任意一种或多种方式发送请求消息：HTTP/2 请求报文的扩展首部的键值对形式和 WebSocket 的帧中的 JSON 键值对形式。

本公开的实施例还提供了另一种存储介质。可选地，在本实施例中，上述存储介质可以被设置为存储用于执行以下步骤的程序代码：

S1，媒体服务器接收客户端发送的用于获取媒体呈现描述文件 MPD 的请求消息，其中，请求信息至少包括：快速启动推送指令和 MPD 请求指令；

S2，媒体服务器根据请求消息生成响应信息，其中，响应消息包括：确定客户端需要快速启动的推送确认指令、响应 MPD 请求指令得到的 MPD 数据；

S3，在媒体服务器生成预测的片段消息之后，媒体服务器将预测的片段消息和响应信息推送至客户端，使得客户端根据预测的片段消息和 MPD 数据，获取客户端需要的媒体内容，其中，预测的片段消息为媒体服务器主动推送的与推送确认指令对应的片段消息。

可选地，在本实施例中，上述存储介质可以包括但不限于：U 盘、只读存储器（ROM，Read-Only Memory）、随机存取存储器（RAM，Random Access Memory）、移动硬盘、磁碟或者光盘等各种可以存储程序代码的介质。

可选地，本实施例中的具体示例可以参考上述实施例及可选实施方式中所描述的示例，本实施例在此不再赘述。

显然，本领域的技术人员应该明白，上述的本公开的各模块或各步骤可以用通用的计算装置来实现，它们可以集中在单个的计算装置上，或者分布在多个计算装置所组成的网络上，可选地，它们可以用计算装置可执行的程序代码来实现，从而，可以将它们存储在存储装置中由计算装置来执行，并且在某些情况下，可以以不同于此处的顺序执行所示出或描述的步骤，或者将它们分别制作成各个集成电路模块，或者将它们中的多个模块或步骤制作成单个集成电路模块来实现。这样，本公开不限制于任何特定的硬件和软件结合。

以上所述仅为本公开的优选实施例而已，并不用于限制本公开，对于本领域的技术人员来说，本公开可以有各种更改和变化。凡在本公开的精神和原则之内，所作的任何修改、等同替换、改进等，均应包含在本公开的保护范围之内。

工业实用性：

本公开涉及流媒体领域，提供了一种流媒体快速启动方法、装置和系统，采用客户端通过生成所述 **fast-start** 类型推送指令，即可告诉媒体服务器，客户端期待媒体服务器推送片段序列，以快速启动；通过使用从所述媒体服务器接收服务器推送的  $K$  个片段的片段消息，减少了客户端盲目等待的时间，降低了启动延迟，解决了媒体客户端与媒体服务器端之间没有达到很好的协商同步的问题，达到了媒体客户端可以判断自己期望的资源是否在推送列表，以便尽早做出下一步决策，实现更好的快速启动的效果。

## 权利要求书

1. 一种流媒体快速启动方法，包括：

客户端向媒体服务器发送快速启动类型推送指令，其中，所述快速启动类型推送指令用于指示所述媒体服务器推送初始化片段序列和/或媒体片段序列；

所述客户端接收所述媒体服务器返回的快速启动类型推送确认指令；

所述客户端根据所述快速启动类型推送确认指令获取所述媒体服务器将要推送的初始化片段和/或媒体片段序列的信息。

2. 根据权利要求 1 所述的方法，其中，客户端向媒体服务器发送快速启动类型推送指令包括：

所述客户端通过 MPD 请求消息向所述媒体服务器发送所述快速启动类型推送指令。

3. 根据权利要求 2 所述的方法，其中，所述客户端接收所述快速启动类型推送确认指令包括：

所述客户端接收所述媒体服务器返回的 MPD 响应消息，所述 MPD 响应消息至少包含 MPD 数据，以及所述快速启动类型推送确认指令。

4. 根据权利要求 1 或 3 所述的方法，其中，所述客户端根据所述快速启动类型推送确认指令获取所述服务器将要推送的初始化片段和/或媒体片段序列的信息，包括：

所述快速启动类型推送确认指令中携带 URL 模板类型的推送参数；

所述客户端解析 URL 模板推送参数，获取所述媒体服务器将要推送的初始化片段序列和/或媒体片段序列的信息。

5. 根据权利要求 1 或 3 所述的方法，其中，所述客户端根据所述快速启动类型推送确认指令获取所述服务器将要推送的初始化片段和/或媒体片段序列的信息，包括：

所述快速启动类型推送确认指令中携带数值类型的推送参数；

所述客户端解析数值推送参数 K，并接收后续的 K 个推送承诺 PUSH\_PROMISE 帧，获取所述媒体服务器将要推送的初始化片段序列和/或媒体片段序列的信息。

6. 根据权利要求 1 或 3 所述的方法，其中，所述客户端根据所述快速启动类型推送确认指令获取所述服务器将要推送的初始化片段和/或媒体片段序列的信息，包括：

所述快速启动类型推送确认指令中携带 URL 列表类型的推送参数；

所述客户端解析 URL 列表推送参数，获取所述媒体服务器将要推送的初始化片段序列和/或媒体片段序列的信息。

7. 根据权利要求 3 所述的方法，其中，所述方法还包括：

所述客户端解析所述 MPD 数据，确定需要所述媒体服务器推送的初始化片段和/或媒体片段序列的信息；

若获取的媒体服务器将要推送的初始化片段和/或媒体片段序列的信息等同于确定的需要媒体服务器推送的初始化片段和/或媒体片段序列的信息，所述客户端接收所述媒体服务器推送的初始化片段序列和/或媒体片段序列。

8. 根据权利要求 3 所述的方法，其中，所述方法还包括：

所述客户端解析所述 MPD 数据，确定需要所述媒体服务器推送的初始化片段和/或媒体片段序列的信息；

若获取的媒体服务器将要推送的初始化片段和/或媒体片段序列的信息不同于确定的需要媒体服务器推送的初始化片段和/或媒体片段序列的信息，所述客户端取消所述媒体服务器推送的初始化片段序列和/或媒体片段序列。

9. 根据权利要求 8 所述的方法，其中，所述方法还包括：

所述客户端向所述媒体服务器发送获取所述确定需要所述媒体服务器推送的初始化片段和/或媒体片段序列的请求。

10. 根据权利要求 1 至 3 中任意一项所述的方法，其中，MPD 请求消息还包括：所述 MPD 请求消息中携带的快速启动类型推送指令包含偏好参数信息，所述偏好参数包括如下任意一种或多种属性：分辨率范围、码率范围、媒体属性和语言。

11. 根据权利要求 1 至 3 中任意一项所述的方法，其中，所述客户端接收所述媒体服务器返回的 MPD 响应消息，以及所述快速启动类型推送确认指令，进一步包括：

通过 HTTP/2 报文方式接收所述 MPD 响应消息，其中，所述 HTTP/2 报文的 Accept-Push-Policy 首部携带快速启动类型推送确认指令；或者，

通过 WebSocket 中 DASH 子协议的新新\_mpd 消息接收所述 MPD 响应消息，其中，所述 new\_mpd 消息的 JSON 键值对携带快速启动类型推送确认指令。

12. 根据权利要求 1 至 3 中任意一项所述的方法，其中，所述客户端向所述媒体服务器发送 MPD 请求消息，以及所述快速启动类型推送指令，进一步包括：

通过 HTTP/2 报文方式发送所述 MPD 请求消息，其中，所述 HTTP/2 报文的 Accept-Push-Policy 首部携带快速启动类型推送指令；或者，

通过 WebSocket 中 DASH 子协议的 get\_mpd 消息发送所述 MPD 请求消息，其中，所述 get\_mpd 消息的 JSON 键值对携带快速启动类型推送确认指令。

13. 一种流媒体快速启动方法，包括：

媒体服务器接收客户端发送的快速启动类型推送指令，其中，所述快速启动类型推

送指令用于指示所述媒体服务器推送初始化片段序列和/或媒体片段序列；

所述媒体服务器根据将要推送的初始化片段和/或媒体片段序列，生成快速启动类型推送确认指令；

所述媒体服务器向所述客户端发送所述快速启动类型推送确认指令。

14. 根据权利要求 13 所述的方法，其中，媒体服务器接收快速启动类型推送指令包括：

所述媒体服务器接收所述客户端发送的 MPD 请求消息，所述 MPD 请求消息中包含所述快速启动类型推送指令。

15. 根据权利要求 13 所述的方法，其中，所述媒体服务器生成快速启动类型推送确认指令，包括：

所述媒体服务器根据将要推送的初始化片段序列和/或媒体片段序列，生成 URL 模板类型的推送参数；

所述快速启动类型推送确认指令中携带所述 URL 模板推送参数。

16. 根据权利要求 13 所述的方法，其中，所述媒体服务器生成快速启动类型推送确认指令，包括：

所述媒体服务器根据将要推送的初始化片段序列和/或媒体片段序列，生成数值类型的推送参数；

所述快速启动类型推送确认指令中携带所述数值类型的推送参数。

17. 根据权利要求 13 所述的方法，其中，所述媒体服务器生成快速启动类型推送确认指令，包括：

所述媒体服务器根据将要推送的初始化片段序列和/或媒体片段序列，生成 URL 列表类型的推送参数；

所述快速启动类型推送确认指令中携带所述 URL 列表推送参数。

18. 根据权利要求 13 至 17 中任意一项所述的方法，其中，发送所述快速启动类型推送确认指令包括：

所述媒体服务器通过 MPD 响应消息向所述客户端发送所述快速启动类型推送确认指令，其中，所述 MPD 响应消息至少包含 MPD 数据，以及所述快速启动类型推送确认指令。

19. 根据权利要求 15 至 17 中任意一项所述的方法，其中，所述媒体服务器生成快速启动类型推送确认指令，包括：

所述媒体服务器根据所述客户端的预定数量和/或所述客户端的历史访问结果，确定

所述快速启动类型推送确认指令中携带的推送参数。

20. 根据权利要求 19 所述的方法，其中，所述媒体服务器向所述客户端发送所述快速启动类型推送确认指令之后，所述方法还包括：

所述媒体服务器向所述客户端推送所述快速启动类型推送确认指令指示的片段序列初始化片段和/或媒体片段序列。

21. 根据权利要求 13 至 17 中任意一项所述的方法，其中，MPD 请求消息还包括：所述 MPD 请求消息中携带的快速启动类型推送指令包含偏好参数信息，所述偏好参数包括如下任意一种或多种属性：分辨率范围、码率范围、媒体属性和语言。

22. 根据权利要求 13 至 17 中任意一项所述的方法，其中，所述媒体服务器接收所述客户端发送的 MPD 请求消息，以及所述快速启动类型推送指令，进一步包括：

通过 HTTP/2 报文方式接收所述 MPD 请求消息，其中，所述 HTTP/2 报文的 Accept-Push-Policy 首部携带快速启动类型推送指令；或者，

通过 WebSocket 中 DASH 子协议的 get\_mpd 消息接收所述 MPD 请求消息，其中，所述 get\_mpd 消息的 JSON 键值对携带快速启动类型推送确认指令。

23. 根据权利要求 13 至 17 中任意一项所述的方法，其中，所述媒体服务器向所述客户端发送 MPD 响应消息，以及所述快速启动类型推送确认指令，进一步包括：

通过 HTTP/2 报文方式发送所述 MPD 响应消息，其中，所述 HTTP/2 报文的 Accept-Push-Policy 首部携带快速启动类型推送确认指令；或者，

通过 WebSocket 中 DASH 子协议的新 new\_mpd 消息发送所述 MPD 响应消息，其中，所述 new\_mpd 消息的 JSON 键值对携带快速启动类型推送确认指令。

24. 一种流媒体快速启动装置，包括：

访问引擎模块，设置为客户端向媒体服务器发送快速启动类型推送指令，其中，所述快速启动类型推送指令用于指示所述媒体服务器推送初始化片段序列和/或媒体片段序列；

第一接收模块，设置为所述客户端接收所述媒体服务器返回的快速启动类型推送确认指令；

控制引擎模块，设置为所述客户端根据所述快速启动类型推送确认指令获取所述媒体服务器将要推送的初始化片段和/或媒体片段序列的信息。

25. 根据权利要求 24 所述的装置，其中，所述访问引擎模块包括，：

发送模块，设置为所述客户端通过 MPD 请求消息向所述媒体服务器发送所述快速启动类型推送指令。

26. 根据权利要求 25 所述的装置，其中，所述第一接收模块还包括：

第一接收子模块，设置为所述客户端接收所述媒体服务器返回的 MPD 响应消息，所述 MPD 响应消息至少包含 MPD 数据，以及所述快速启动类型推送确认指令。

27. 根据权利要求 24 或 26 所述的装置，其中，所述控制引擎模块包括：

第一确认模块，设置为所述快速启动类型推送确认指令中携带 URL 模板 URL\_TEMPLATE 类型的推送参数；

第一获取模块，设置为所述客户端解析 URL 模板推送参数，获取所述媒体服务器将要推送的初始化片段序列和/或媒体片段序列的信息。

28. 根据权利要求 24 或 26 所述的装置，其中，所述控制引擎模块包括：

第二确认模块，设置为所述快速启动类型推送确认指令中携带数值 NUMBER 类型的推送参数；

第二获取模块，设置为所述客户端解析数值推送参数 K，并接收后续的 K 个推送承诺 PUSH\_PROMISE 帧，获取所述媒体服务器将要推送的初始化片段序列和/或媒体片段序列的信息。

29. 根据权利要求 24 或 26 所述的装置，其中，所述控制引擎模块包括：

第三确认模块，设置为所述快速启动类型推送确认指令中携带 URL 列表 URL\_LIST 类型的推送参数；

第三获取模块，设置为所述客户端解析 URL 列表推送参数，获取所述媒体服务器将要推送的初始化片段序列和/或媒体片段序列的信息。

30. 根据权利要求 26 中任意一项所述的装置，其中，所述装置还包括：

第四确认模块，设置为所述客户端解析所述 MPD 数据，确定需要所述媒体服务器推送的初始化片段和/或媒体片段序列的信息；

第二接收模块，设置为若获取的媒体服务器将要推送的初始化片段和/或媒体片段序列的信息等同于确定的需要媒体服务器推送的初始化片段和/或媒体片段序列的信息，所述客户端接收所述媒体服务器推送的初始化片段序列和/或媒体片段序列。

31. 一种流媒体快速启动装置，包括：

服务模块，设置为媒体服务器接收客户端发送的快速启动类型推送指令；

生成引擎模块，设置为所述媒体服务器根据将要推送的初始化片段和/或媒体片段序列，生成快速启动类型推送确认指令；

第一推送模块，设置为所述媒体服务器向所述客户端发送所述快速启动类型推送确

认指令。

32. 一种流媒体快速启动系统，其中，包括：

客户端，设置为向媒体服务器发送所述快速启动类型推送指令，并接收媒体服务器返回的快速启动类型推送确认指令，根据所述快速启动类型推送确认指令获取所述服务器将要推送的初始化片段和/或媒体片段序列的信息，其中，所述快速启动类型推送指令用于指示所述媒体服务器推送初始化片段序列和/或媒体片段序列；

所述媒体服务器，与所述客户端通信，设置为接收所述客户端发送的所述快速启动类型推送指令，生成快速启动类型推送确认指令，向所述客户端发送所述快速启动类型推送确认指令，并向所述客户端推送所述初始化片段序列和/或所述媒体片段序列。

33. 一种存储介质，所述存储介质包括存储的程序，其中，所述程序运行时执行权利要求 1 至 24 中任一项所述的方法。

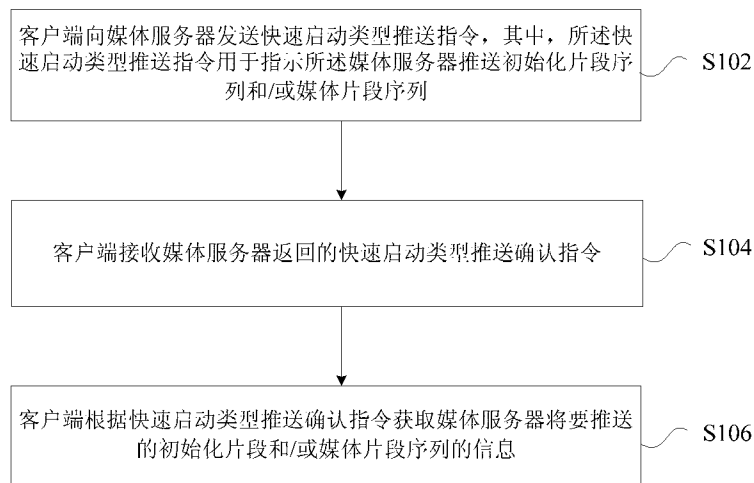


图 1

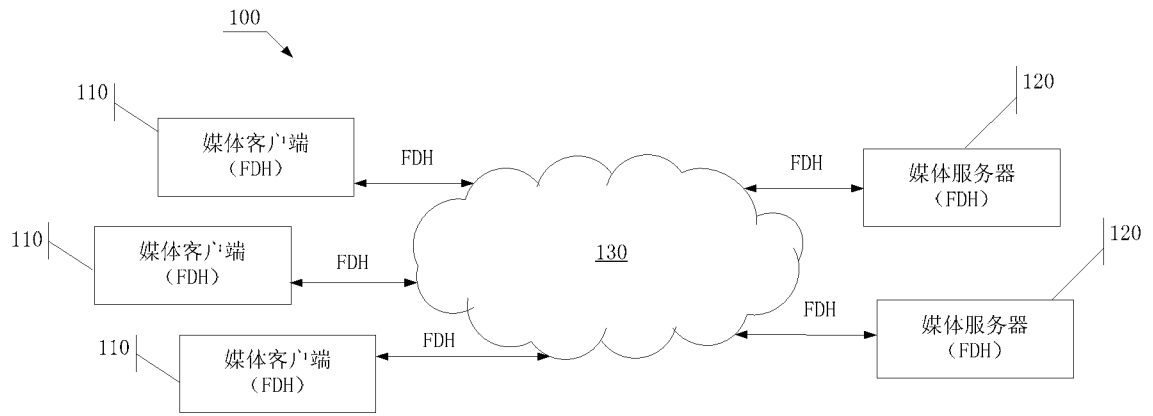


图 2

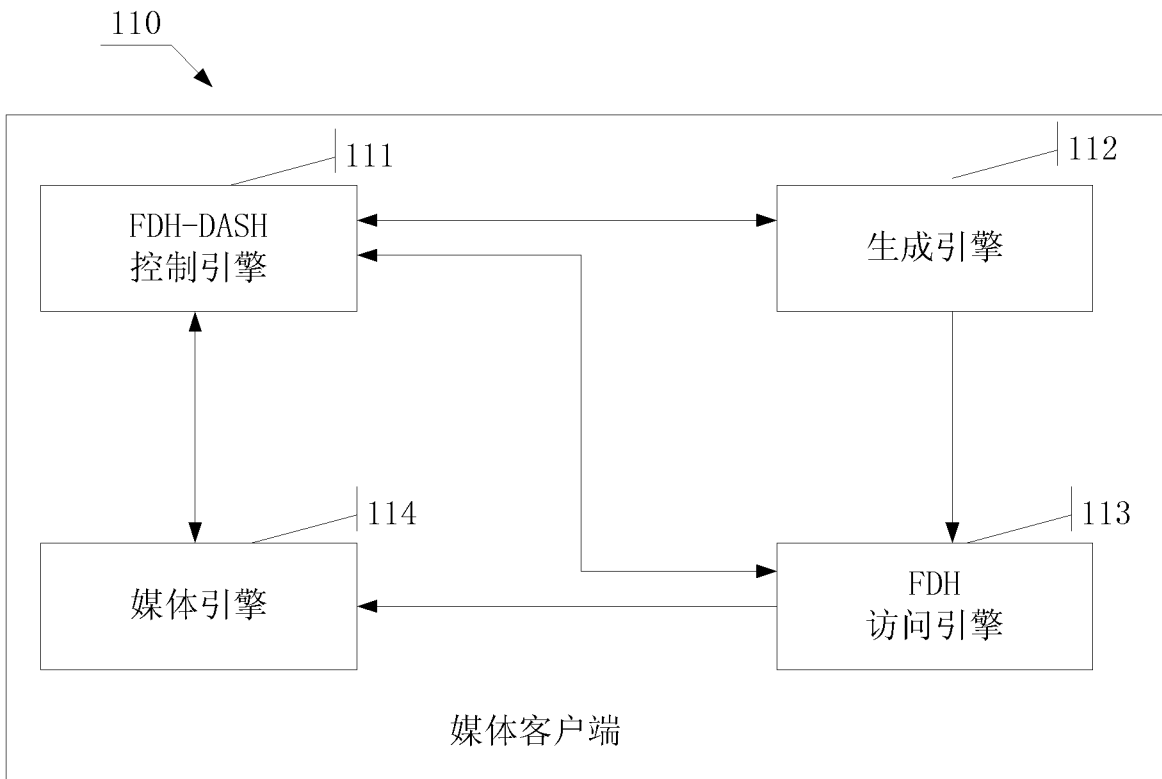


图 3

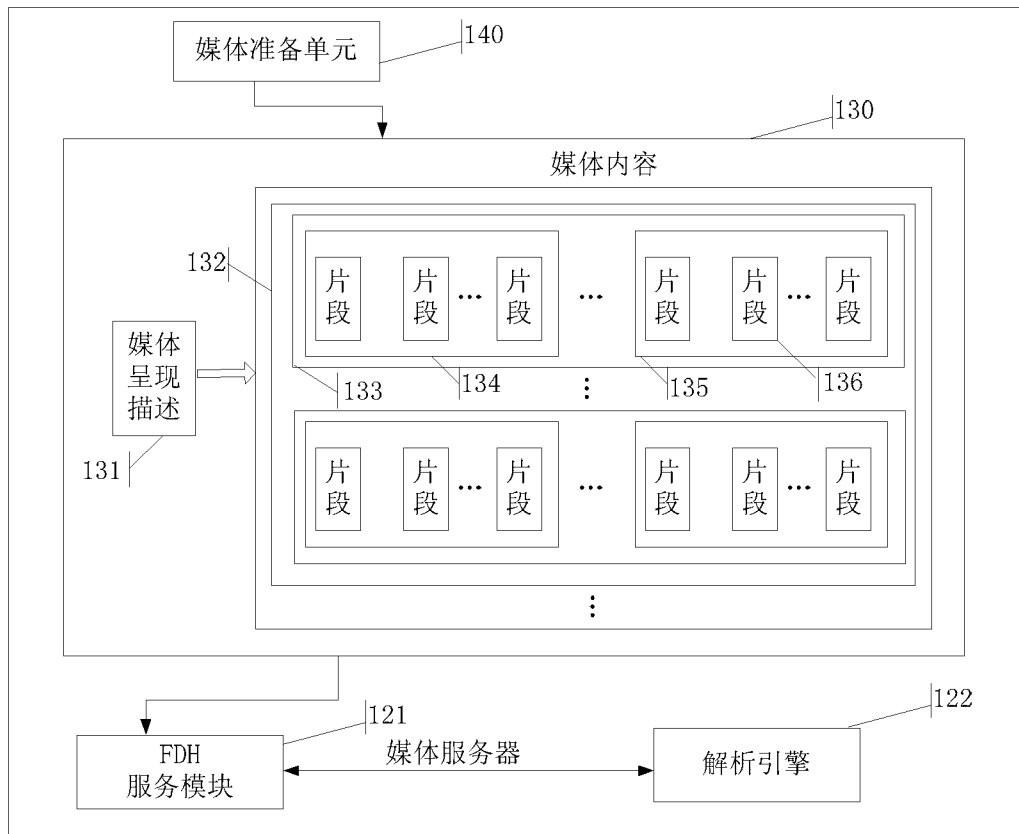


图 4

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- MPD file generated with GPP version 0.5.1-RC1-999579 on 2014-03-04T10:57:02Z -->
<MPD xmlns="urn:mpeg:dash:schema:mpd:2011" minBufferTime="PT1.500000S" type="static" mediaPresentationDuration="PT0H0M2.795S" profiles="urn:mpeg:dash:profile:isoff-live:2011">
  <!--dash:EPheasantFrom_Is_sample_2014_05_09.mpd generated by GPPC/Tiles-->
  <ProgramInformation>
    <Title>dash/EPheasantFrom_Is_sample_2014_05_09.mpd generated by GPPC/Tiles</Title>
  </ProgramInformation>
  <Period duration="PT0H0M2.795S">
    <AdaptationSet segmentAlignment="vr" group="1" maxBitrate="460" maxFrameRate="360" maxFrameRate="24" par="4,3">
      <Representation timescale="96" media="ed:StandardH264Sps/EPheasantFrom_IsSample$mas_startNumber$mas_startNumber+1" duration="96" initialization="ed:StandardH264Sps/EPheasantFrom_Is_init.mpd" />
      <Representation id="320x240 48.0Mbps" mimeType="video/mp4" codecs="avc1.42c014" width="320" height="240" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="47709" />
      <Representation id="320x240 59.0Mbps" mimeType="video/mp4" codecs="avc1.42c014" width="320" height="240" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="92690" />
      <Representation id="320x240 135.0Mbps" mimeType="video/mp4" codecs="avc1.42c014" width="320" height="240" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="133372" />
      <Representation id="480x360 184.0Mbps" mimeType="video/mp4" codecs="avc1.42c015" width="480" height="360" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="183892" />
      <Representation id="480x360 228.0Mbps" mimeType="video/mp4" codecs="avc1.42c015" width="480" height="360" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="228071" />
      <Representation id="480x360 270.0Mbps" mimeType="video/mp4" codecs="avc1.42c015" width="480" height="360" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="344976" />
      <Representation id="480x360 406.0Mbps" mimeType="video/mp4" codecs="avc1.42c015" width="480" height="360" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="405893" />
      <Representation id="854x480 542.0Mbps" mimeType="video/mp4" codecs="avc1.42c01e" width="854" height="480" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="541708" />
      <Representation id="854x480 618.0Mbps" mimeType="video/mp4" codecs="avc1.42c01e" width="854" height="480" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="617663" />
      <Representation id="1280x720 808.0Mbps" mimeType="video/mp4" codecs="avc1.42c01f" width="1280" height="720" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="808384" />
      <Representation id="1280x720 1.10Mbps" mimeType="video/mp4" codecs="avc1.42c01f" width="1280" height="720" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="1055705" />
      <Representation id="1280x720 1.30Mbps" mimeType="video/mp4" codecs="avc1.42c01f" width="1280" height="720" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="1273567" />
      <Representation id="1280x720 1.60Mbps" mimeType="video/mp4" codecs="avc1.42c01f" width="1280" height="720" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="1566198" />
      <Representation id="1920x1080 2.30Mbps" mimeType="video/mp4" codecs="avc1.42c032" width="1920" height="1080" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="2166639" />
      <Representation id="1920x1080 2.50Mbps" mimeType="video/mp4" codecs="avc1.42c032" width="1920" height="1080" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="2343554" />
      <Representation id="1920x1080 3.10Mbps" mimeType="video/mp4" codecs="avc1.42c032" width="1920" height="1080" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="3127680" />
      <Representation id="1920x1080 3.60Mbps" mimeType="video/mp4" codecs="avc1.42c032" width="1920" height="1080" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="3603876" />
      <Representation id="1920x1080 4.00Mbps" mimeType="video/mp4" codecs="avc1.42c032" width="1920" height="1080" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="4002334" />
      <Representation id="1920x1080 4.50Mbps" mimeType="video/mp4" codecs="avc1.42c032" width="1920" height="1080" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="4516590" />
    </AdaptationSet>
  </Period>
</MPD>

```

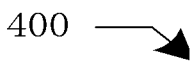


图 5A

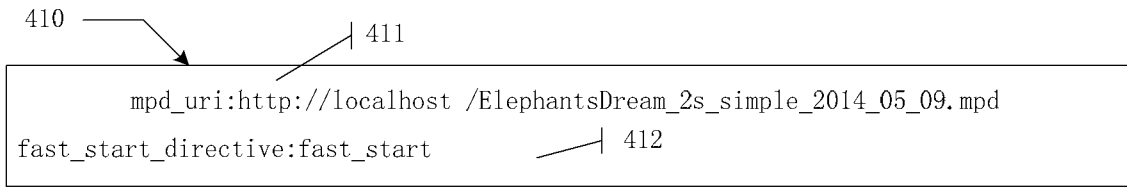


图 5B

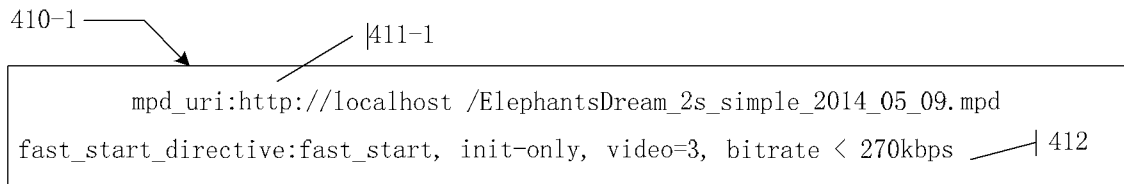


图 5B-1

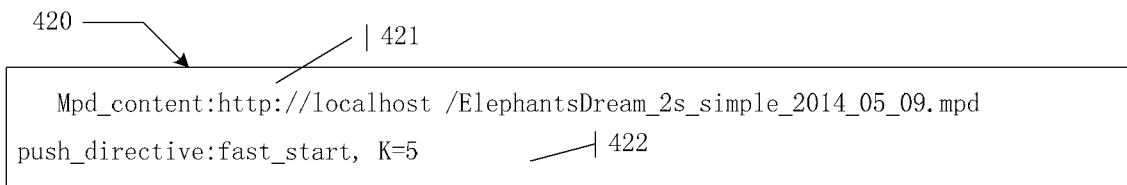


图 5C

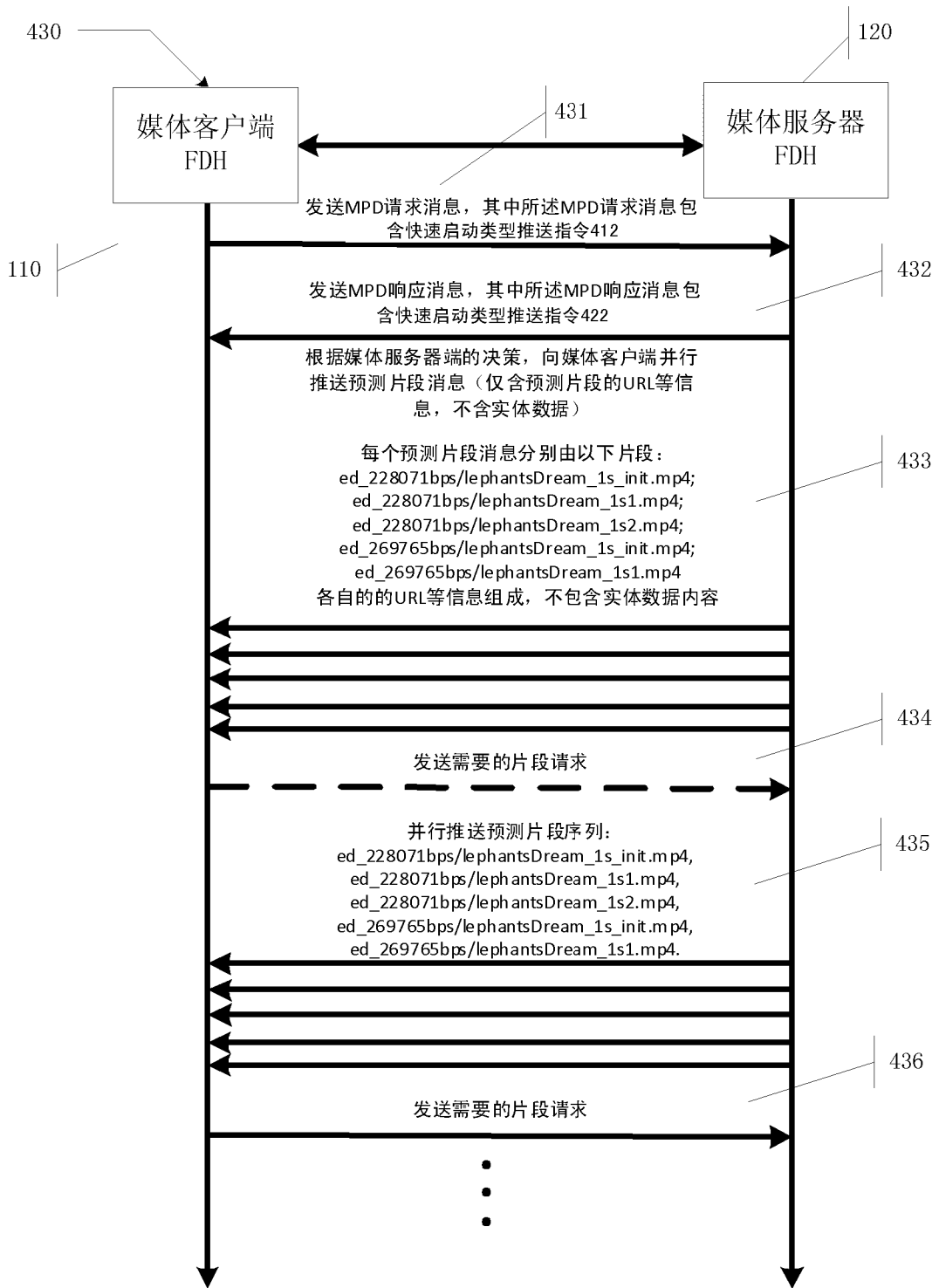


图 5D

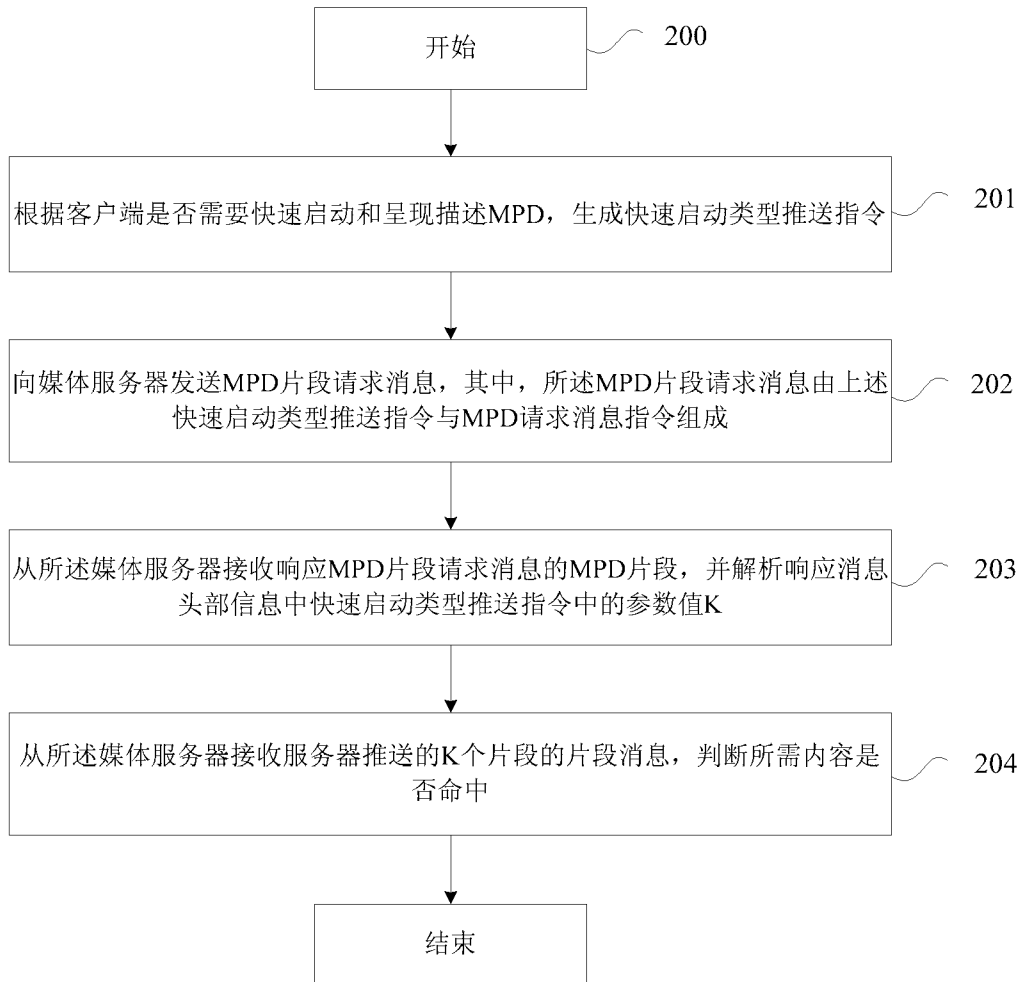


图 6

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- MPD file generated with SPS version 0.5.1-2017-09-03 on 2014-03-09T12:57:02Z -->
<MPD xmlns="urn:mpeg:dash:schema:mpd:2011" minBufferTime="PT.5000000S" type="static" mediaPresentationDuration="PT0H0M2.795S" profiles="urn:mpeg:dash:profile:live:2011">
  <Period start="2014-03-09T12:57:02Z" end="2014-03-09T12:57:02Z">
    <ProgramInformation start="2014-03-09T12:57:02Z" end="2014-03-09T12:57:02Z" title="dashod/RepurposedStream_1s_sample_2014_03_09.mpd generated by GMA-C/11110">
      <URI>dashod/RepurposedStream_1s_sample_2014_03_09.mpd generated by GMA-C/11110</URI>
    </ProgramInformation>
    <Period duration="PT0H0M2.795S">
      <AdaptationSet segmentAlignment="true" group="1" maxBitrate="480" maxHeight="360" maxWidth="640" minBitrate="24" par="1:1">
        <Representation timescale="96" media="ed_SharedAudio/RepurposedStream_1s$Number$_m4s" startByte="1" duration="96" initialization="ed_SharedAudio/RepurposedStream_1s_init.m4s" />
        <Representation id="320x240_48_0kbps" mimeType="video/mp4" codecs="avc1.42c015" width="320" height="240" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="4290" />
        <Representation id="320x240_90_0kbps" mimeType="video/mp4" codecs="avc1.42c015" width="320" height="240" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="8580" />
        <Representation id="320x240_135_0kbps" mimeType="video/mp4" codecs="avc1.42c015" width="320" height="240" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="12870" />
        <Representation id="480x360_184_0kbps" mimeType="video/mp4" codecs="avc1.42c015" width="480" height="360" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="18382" />
        <Representation id="480x360_228_0kbps" mimeType="video/mp4" codecs="avc1.42c015" width="480" height="360" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="22971" />
        <Representation id="480x360_270_0kbps" mimeType="video/mp4" codecs="avc1.42c015" width="480" height="360" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="28560" />
        <Representation id="480x360_345_0kbps" mimeType="video/mp4" codecs="avc1.42c015" width="480" height="360" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="34976" />
        <Representation id="480x360_406_0kbps" mimeType="video/mp4" codecs="avc1.42c015" width="480" height="360" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="40582" />
        <Representation id="854x480_542_0kbps" mimeType="video/mp4" codecs="avc1.42c015" width="854" height="480" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="54178" />
        <Representation id="854x480_618_0kbps" mimeType="video/mp4" codecs="avc1.42c015" width="854" height="480" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="61763" />
        <Representation id="1280x720_808_0kbps" mimeType="video/mp4" codecs="avc1.42c01f" width="1280" height="720" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="80834" />
        <Representation id="1280x720_1_3Mbps" mimeType="video/mp4" codecs="avc1.42c01f" width="1280" height="720" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="105705" />
        <Representation id="1280x720_1_3Mbps" mimeType="video/mp4" codecs="avc1.42c01f" width="1280" height="720" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="127336" />
        <Representation id="1280x720_1_3Mbps" mimeType="video/mp4" codecs="avc1.42c01f" width="1280" height="720" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="156158" />
        <Representation id="1920x1080_2_2Mbps" mimeType="video/mp4" codecs="avc1.42c032" width="1920" height="1080" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="216163" />
        <Representation id="1920x1080_2_2Mbps" mimeType="video/mp4" codecs="avc1.42c032" width="1920" height="1080" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="264354" />
        <Representation id="1920x1080_3_3Mbps" mimeType="video/mp4" codecs="avc1.42c032" width="1920" height="1080" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="312768" />
        <Representation id="1920x1080_3_3Mbps" mimeType="video/mp4" codecs="avc1.42c032" width="1920" height="1080" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="360366" />
        <Representation id="1920x1080_4_4Mbps" mimeType="video/mp4" codecs="avc1.42c032" width="1920" height="1080" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="402234" />
        <Representation id="1920x1080_4_4Mbps" mimeType="video/mp4" codecs="avc1.42c032" width="1920" height="1080" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="451690" />
      </AdaptationSet>
    </Period>
  </MPD>

```

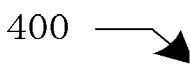


图 7A

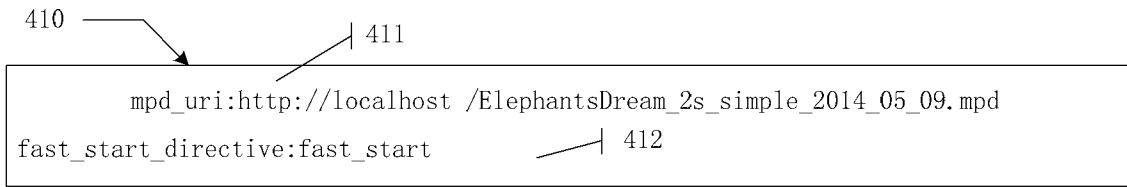


图 7B

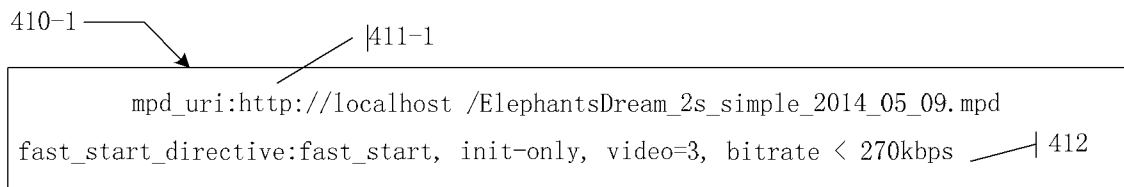


图 7B-1

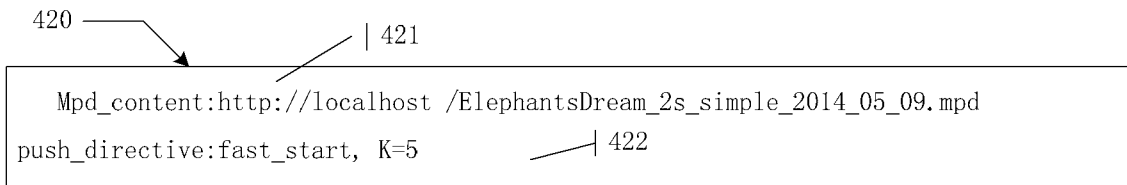


图 7C

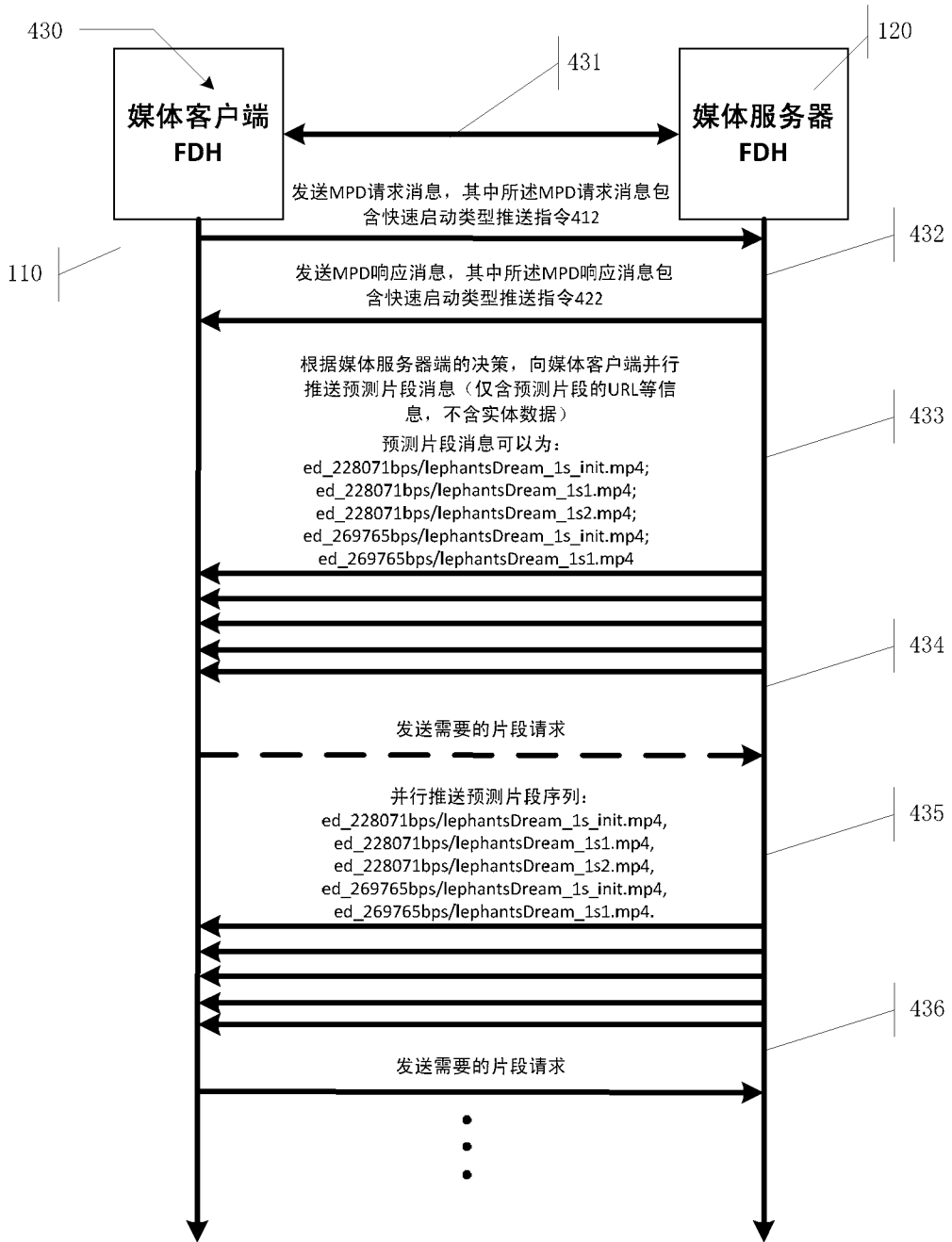


图 7D

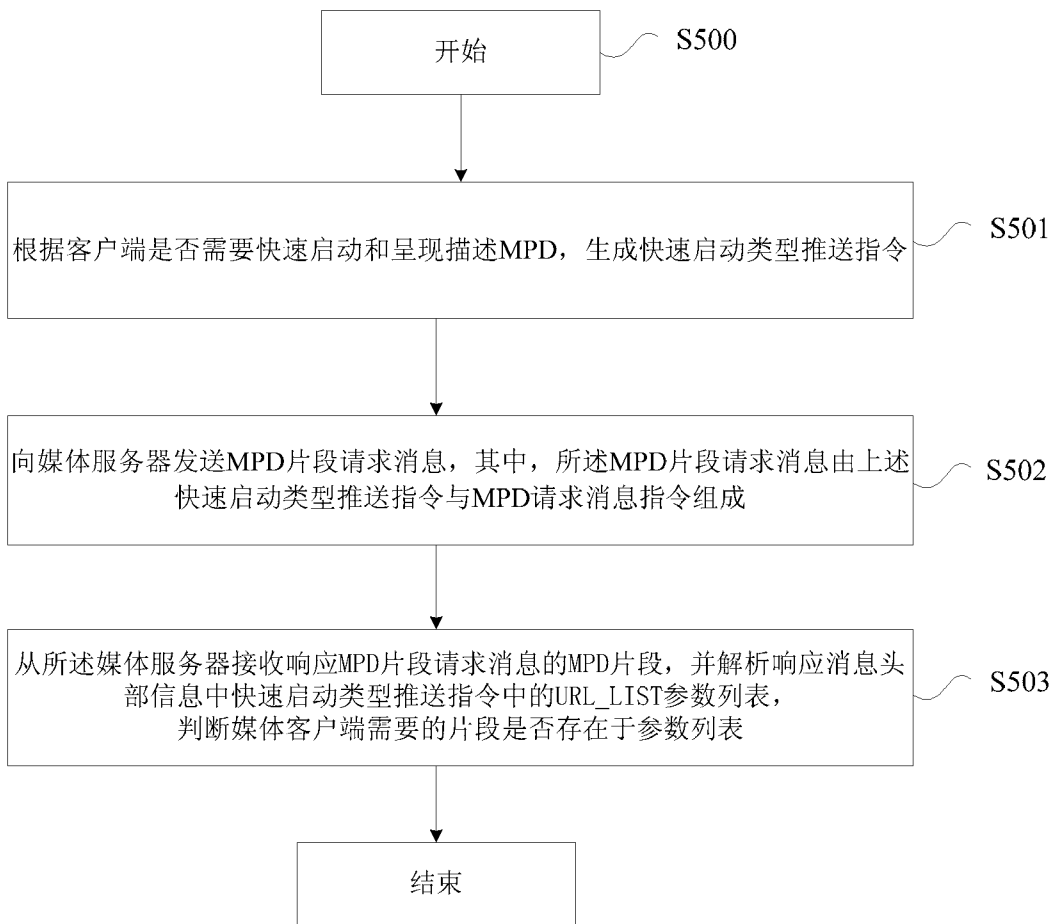


图 8

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- MPD file generated with GMP version 0.5.1-DEV-140319 on 2014-09-09T12:57:02Z -->
<MPD xmlns="urn:mpeg:dash:schema:mpd:2011" minBufferTime="PT1.500000S" type="static" mediaPresentationDuration="PT0H0M2.79S" profiles="urn:mpeg:dash:profile:isoff-live:2011">
  <ProgramInformation xmlns:dash="http://www.dash-project.org/">
    <Title>dash-dash-epm1s01_2014_09_09.mpd generated by GMP</Title>
  </ProgramInformation>
  <Period duration="PT0H0M2.79S">
    <AdaptationSet segmentAlignment="time" group="1" maxBitrate="480" maxHeight="360" maxFrameRate="24" par="4:3">
      <Representation mimeType="video/mp4" timescale="96" media="sd" StandardID="SVC/Profile/Description" ID="SVCProfile" maxSAR="1" duration="96" initialization="sd" Subcode="SVC/Profile/Description" ID="SVCProfile" />
      <Representation id="320x240_48_0Mbps" mimeType="video/mp4" codecs="avc1.42c015" width="320" height="240" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="47709" />
      <Representation id="320x240_96_0Mbps" mimeType="video/mp4" codecs="avc1.42c015" width="320" height="240" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="99690" />
      <Representation id="480x360_144_0Mbps" mimeType="video/mp4" codecs="avc1.42c015" width="480" height="360" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="138372" />
      <Representation id="480x360_184_0Mbps" mimeType="video/mp4" codecs="avc1.42c015" width="480" height="360" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="183892" />
      <Representation id="480x360_228_0Mbps" mimeType="video/mp4" codecs="avc1.42c015" width="480" height="360" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="229071" />
      <Representation id="480x360_345_0Mbps" mimeType="video/mp4" codecs="avc1.42c015" width="480" height="360" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="344978" />
      <Representation id="480x360_406_0Mbps" mimeType="video/mp4" codecs="avc1.42c015" width="480" height="360" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="408692" />
      <Representation id="854x480_542_0Mbps" mimeType="video/mp4" codecs="avc1.42c015" width="854" height="480" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="541708" />
      <Representation id="854x480_618_0Mbps" mimeType="video/mp4" codecs="avc1.42c015" width="854" height="480" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="617603" />
      <Representation id="1280x720_808_0Mbps" mimeType="video/mp4" codecs="avc1.42c015" width="1280" height="720" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="808384" />
      <Representation id="1280x720_1.1Mbps" mimeType="video/mp4" codecs="avc1.42c015" width="1280" height="720" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="1085705" />
      <Representation id="1280x720_1.3Mbps" mimeType="video/mp4" codecs="avc1.42c015" width="1280" height="720" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="1273596" />
      <Representation id="1280x720_1.6Mbps" mimeType="video/mp4" codecs="avc1.42c015" width="1280" height="720" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="1560158" />
      <Representation id="1920x1080_2.3Mbps" mimeType="video/mp4" codecs="avc1.42c032" width="1920" height="1080" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="2186663" />
      <Representation id="1920x1080_2.9Mbps" mimeType="video/mp4" codecs="avc1.42c032" width="1920" height="1080" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="2543554" />
      <Representation id="1920x1080_3.1Mbps" mimeType="video/mp4" codecs="avc1.42c032" width="1920" height="1080" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="3127680" />
      <Representation id="1920x1080_3.6Mbps" mimeType="video/mp4" codecs="avc1.42c032" width="1920" height="1080" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="3603856" />
      <Representation id="1920x1080_4.0Mbps" mimeType="video/mp4" codecs="avc1.42c032" width="1920" height="1080" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="4002334" />
      <Representation id="1920x1080_4.3Mbps" mimeType="video/mp4" codecs="avc1.42c032" width="1920" height="1080" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="4516590" />
    </AdaptationSet>
  </Period>
</MPD>

```

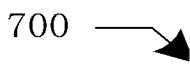


图 9A

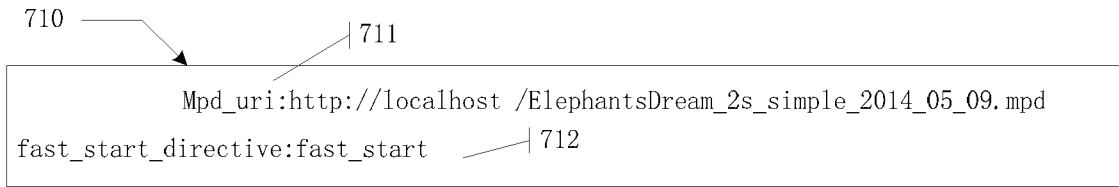


图 9B

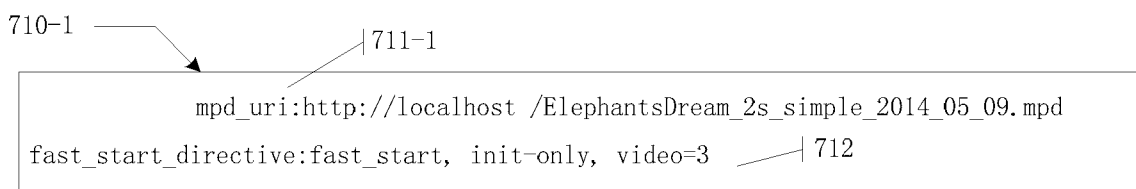


图 9B-1

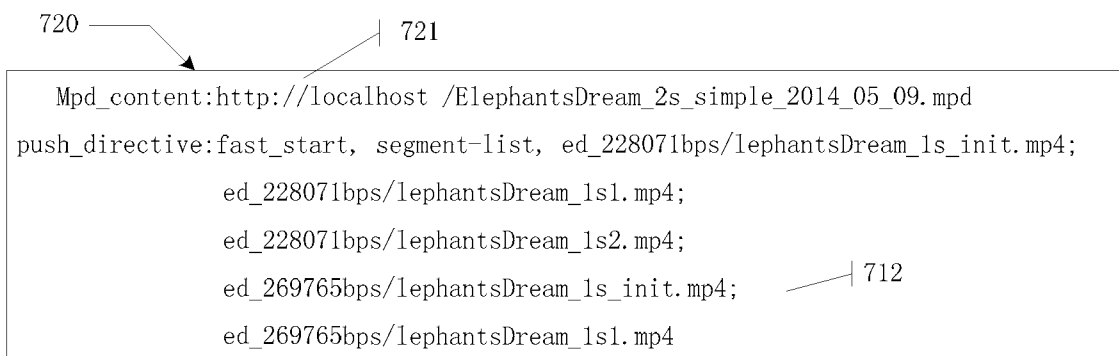


图 9C

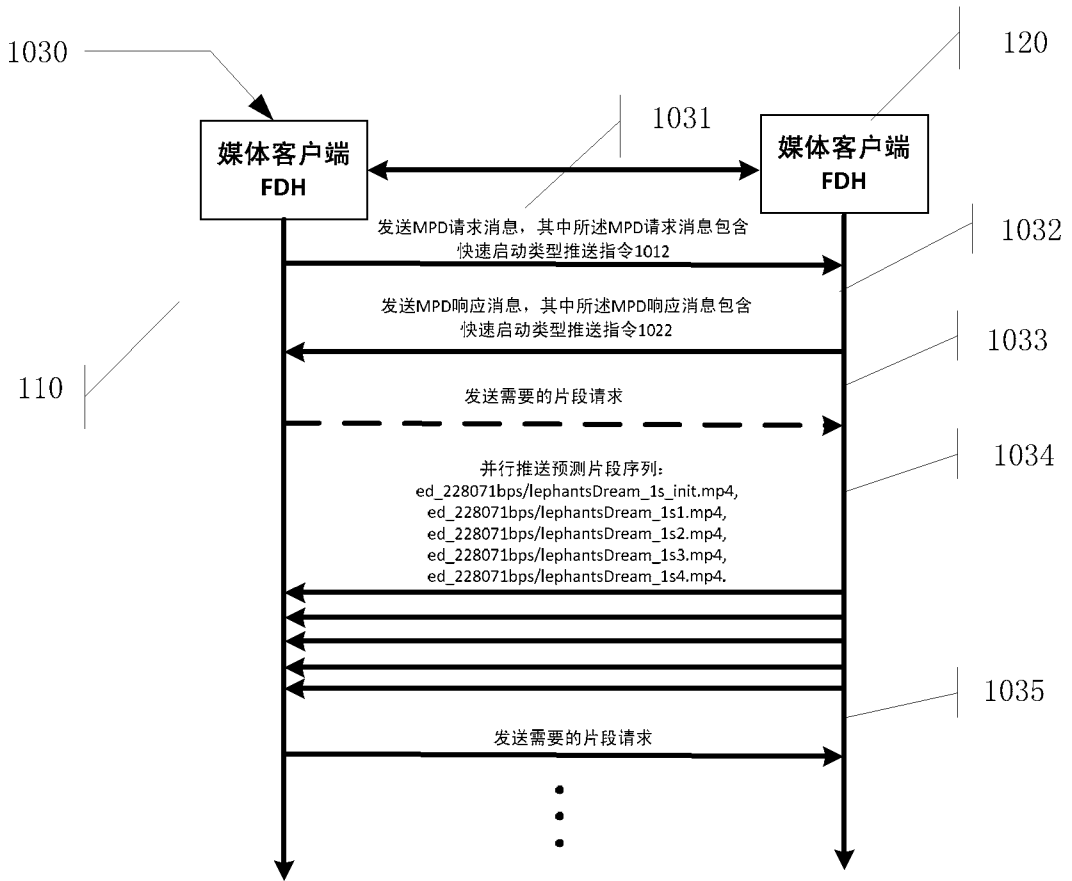


图 9D

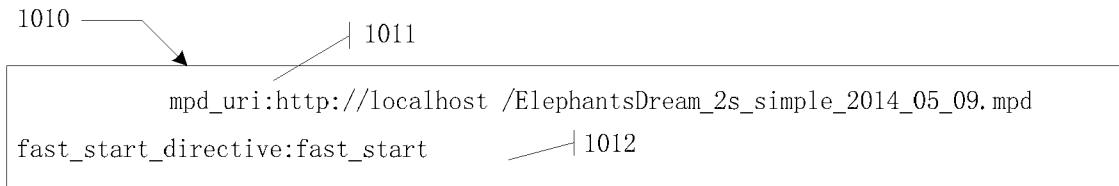
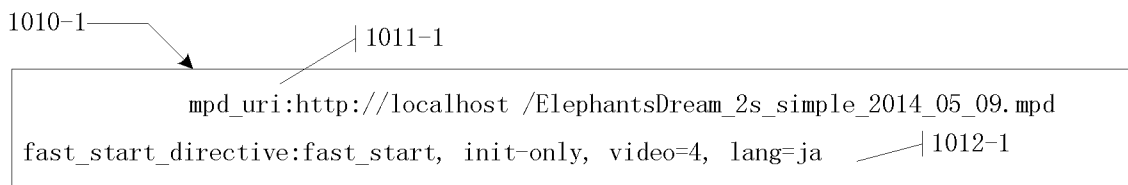
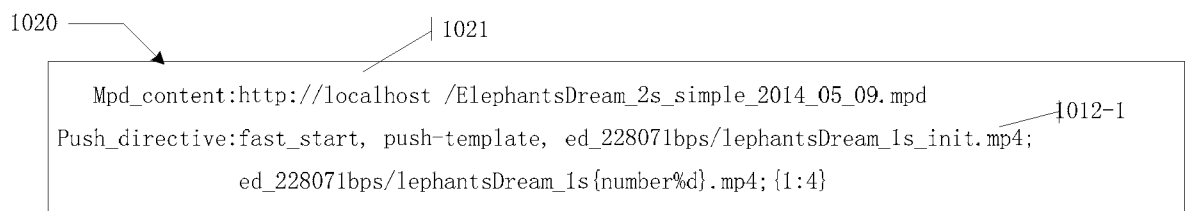
```

<?xml version="1.0" encoding="UTF-8"?>
<!-- MPD file generated with smp version 0.5.4-DEV-c95379 on 2014-09-09T11:57:42Z -->
<MPD xmlns="urn:mpeg:dash:schema:mpd:2011" minBufferTime="PT1.500000S" type="static" mediaPresentationDuration="PT0H0M2.795S" profiles="urn:mpeg:dash:profile:isoff-live:2011">
  <ProgramInformation schemeIdUri="http://gmae.sourceforge.net/">
    <Title>dash/ElementaryStream_1_smp_2014_09_09.mpd generated by GMAE/Title</Title>
  </ProgramInformation>
  <Period duration="PT0H0M2.795S">
    <AdaptationSet segmentAlignment="time" group="1" maxBitrate="480" maxHeight="360" maxWidth="640" par="4:3">
      <SegmentTemplate timescale="96" media="64_SegmentIndex/ElementaryStream_1$Number$_m4s" startNumber="1" duration="96" initialization="64_SegmentIndex/ElementaryStream_1_smp.m4" />
      <Representation id="290x240 48_0Mbps" mimeType="video/mp4" codecs="avc1.42c010" width="290" height="240" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="4709" />
      <Representation id="320x240 50_0Mbps" mimeType="video/mp4" codecs="avc1.42c010" width="320" height="240" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="9660" />
      <Representation id="320x240 135_0Mbps" mimeType="video/mp4" codecs="avc1.42c010" width="320" height="240" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="135372" />
      <Representation id="480x360 184_0Mbps" mimeType="video/mp4" codecs="avc1.42c015" width="480" height="360" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="183852" />
      <Representation id="480x360 228_0Mbps" mimeType="video/mp4" codecs="avc1.42c015" width="480" height="360" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="228071" />
      <Representation id="480x360 345_0Mbps" mimeType="video/mp4" codecs="avc1.42c015" width="480" height="360" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="344976" />
      <Representation id="480x360 406_0Mbps" mimeType="video/mp4" codecs="avc1.42c015" width="480" height="360" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="405892" />
      <Representation id="854x480 542_0Mbps" mimeType="video/mp4" codecs="avc1.42c01e" width="854" height="480" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="941708" />
      <Representation id="854x480 618_0Mbps" mimeType="video/mp4" codecs="avc1.42c01e" width="854" height="480" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="617603" />
      <Representation id="1280x720 808_0Mbps" mimeType="video/mp4" codecs="avc1.42c01f" width="1280" height="720" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="808384" />
      <Representation id="1280x720 1.1Mbps" mimeType="video/mp4" codecs="avc1.42c01f" width="1280" height="720" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="1057505" />
      <Representation id="1280x720 1.3Mbps" mimeType="video/mp4" codecs="avc1.42c01f" width="1280" height="720" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="1273596" />
      <Representation id="1280x720 1.8Mbps" mimeType="video/mp4" codecs="avc1.42c01f" width="1280" height="720" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="1568158" />
      <Representation id="1920x1080 2.2Mbps" mimeType="video/mp4" codecs="avc1.42c022" width="1920" height="1080" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="2106563" />
      <Representation id="1920x1080 2.9Mbps" mimeType="video/mp4" codecs="avc1.42c022" width="1920" height="1080" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="2543554" />
      <Representation id="1920x1080 3.1Mbps" mimeType="video/mp4" codecs="avc1.42c022" width="1920" height="1080" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="3127680" />
      <Representation id="1920x1080 3.8Mbps" mimeType="video/mp4" codecs="avc1.42c022" width="1920" height="1080" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="3603856" />
      <Representation id="1920x1080 4.0Mbps" mimeType="video/mp4" codecs="avc1.42c022" width="1920" height="1080" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="4002334" />
      <Representation id="1920x1080 4.5Mbps" mimeType="video/mp4" codecs="avc1.42c022" width="1920" height="1080" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="4516590" />
    </AdaptationSet>
  </Period>
</MPD>

```

1000

图 10A

**图 10B****图 10B-1****图 10C**

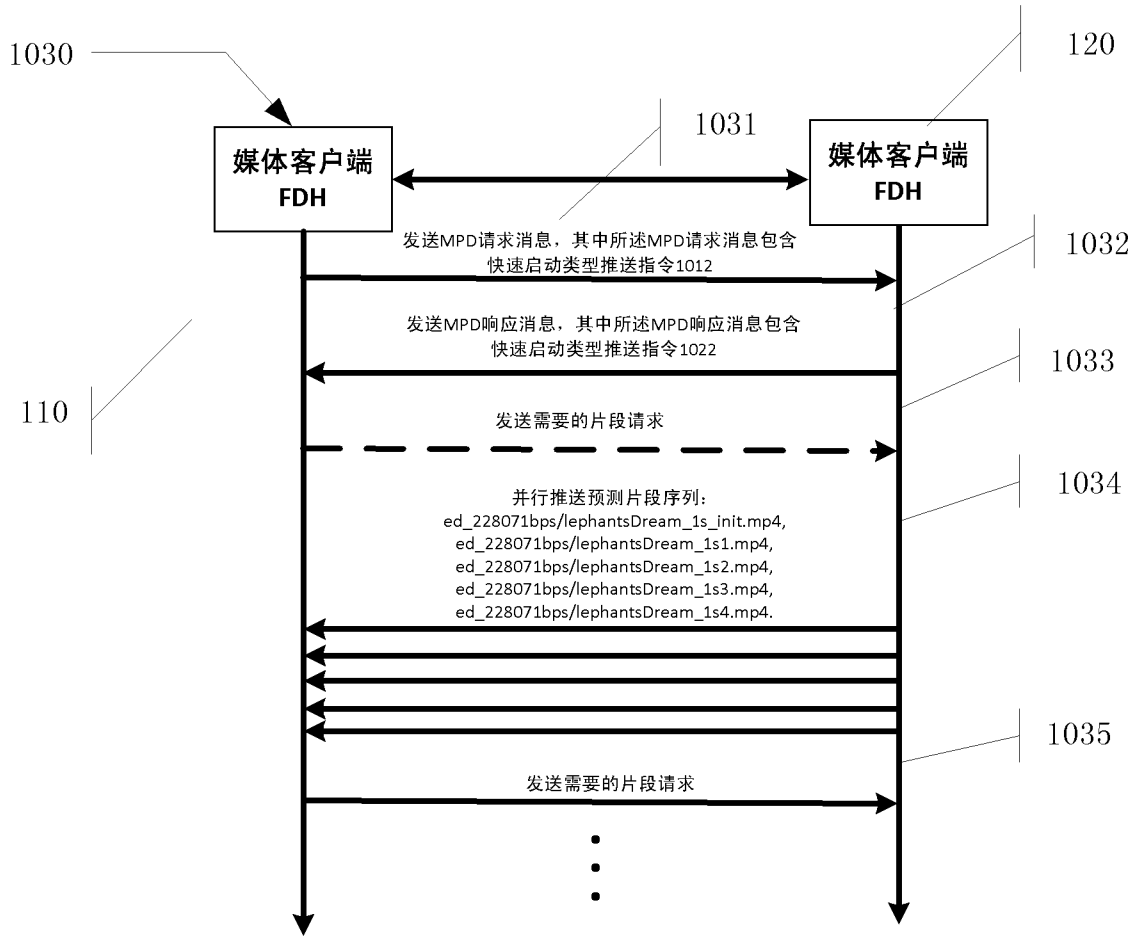


图 10D

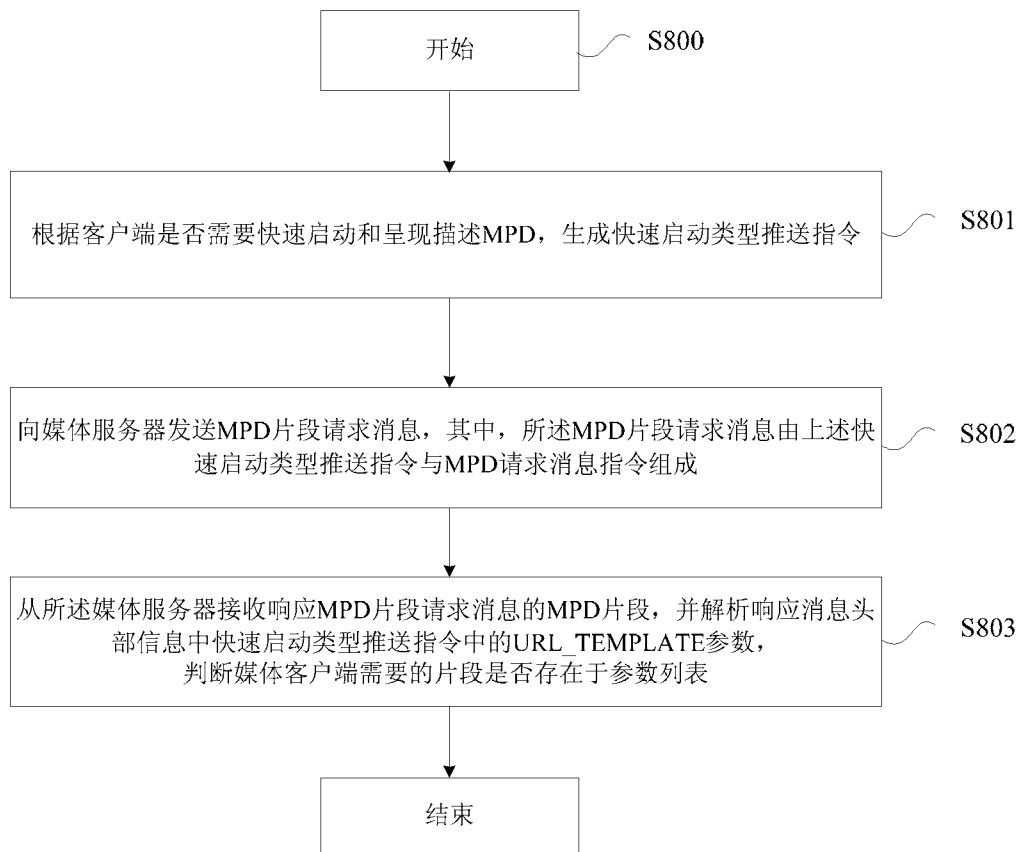


图 11

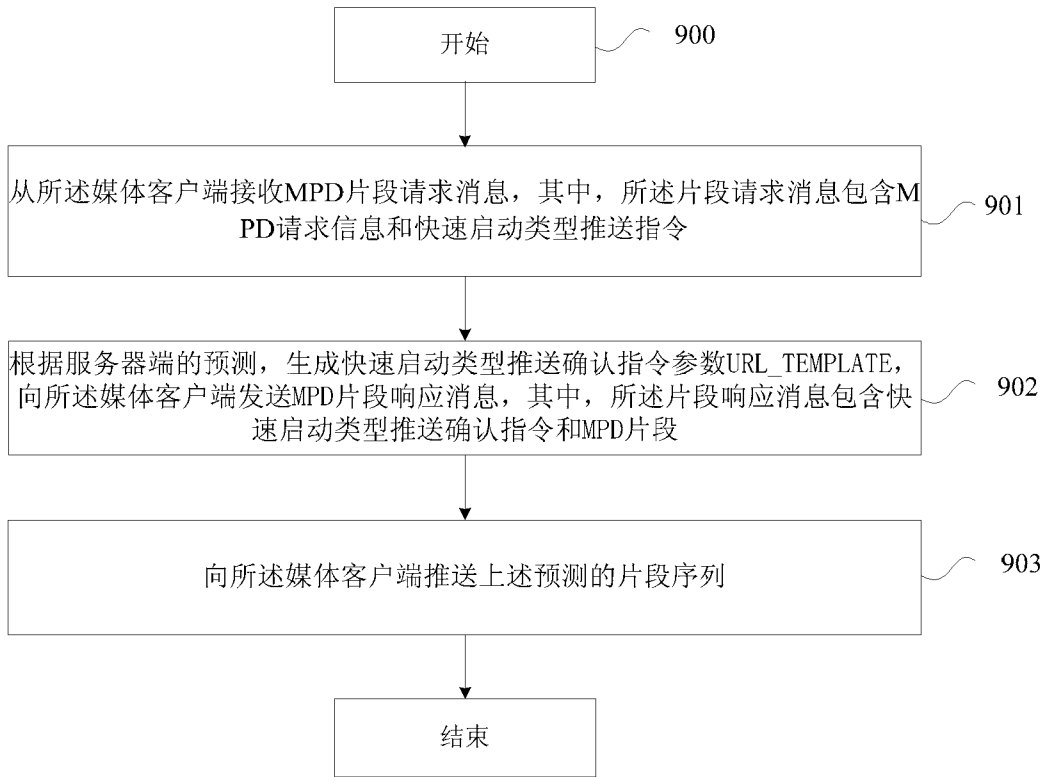


图 12

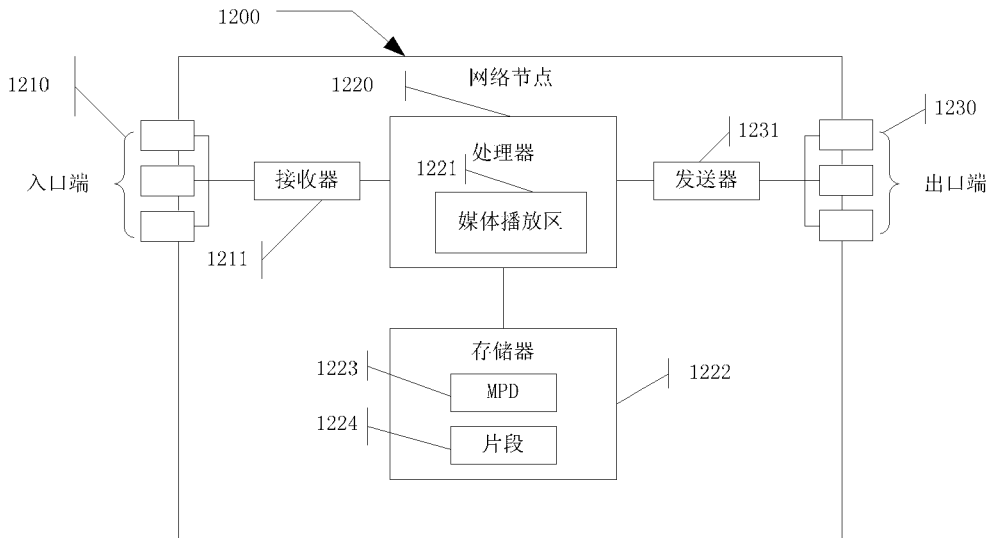


图 13

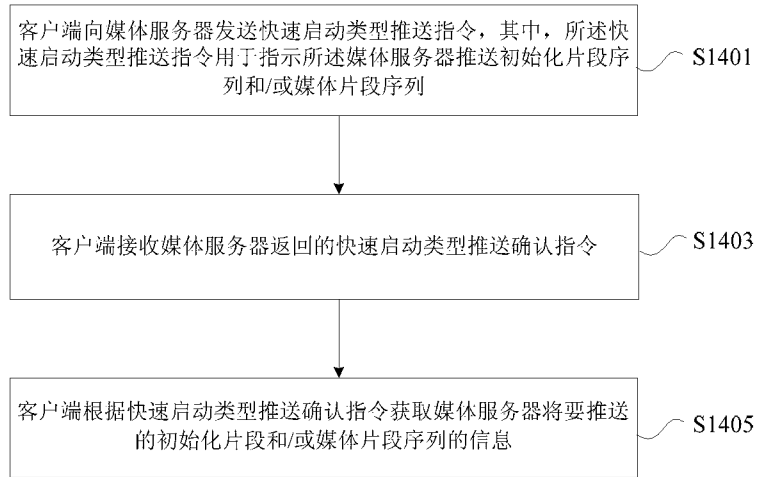


图 14

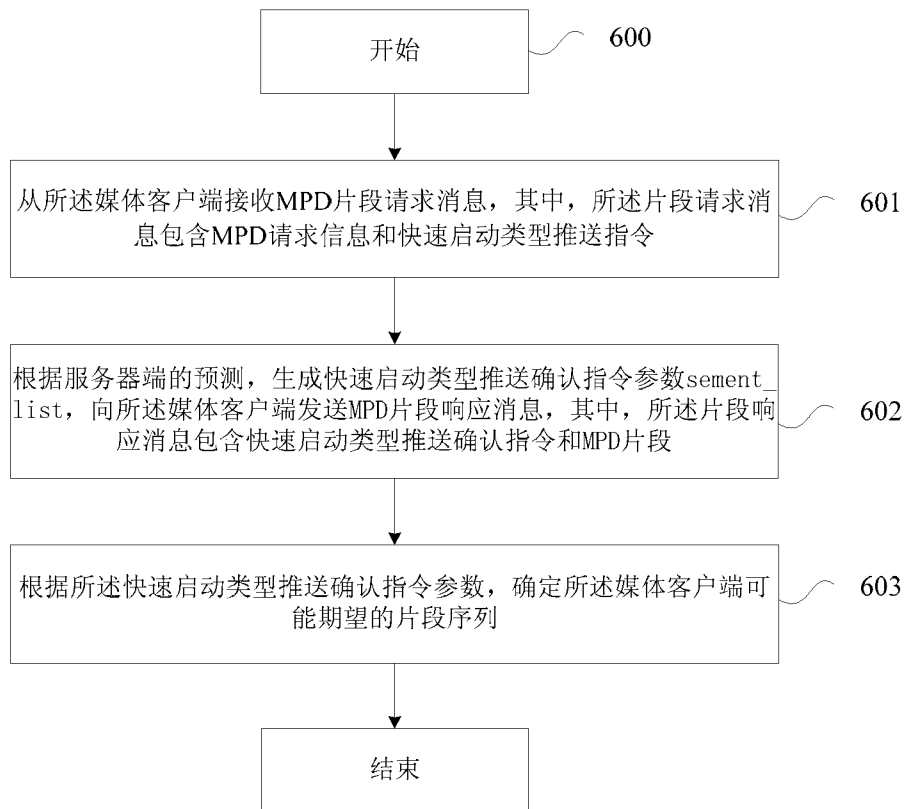


图 15

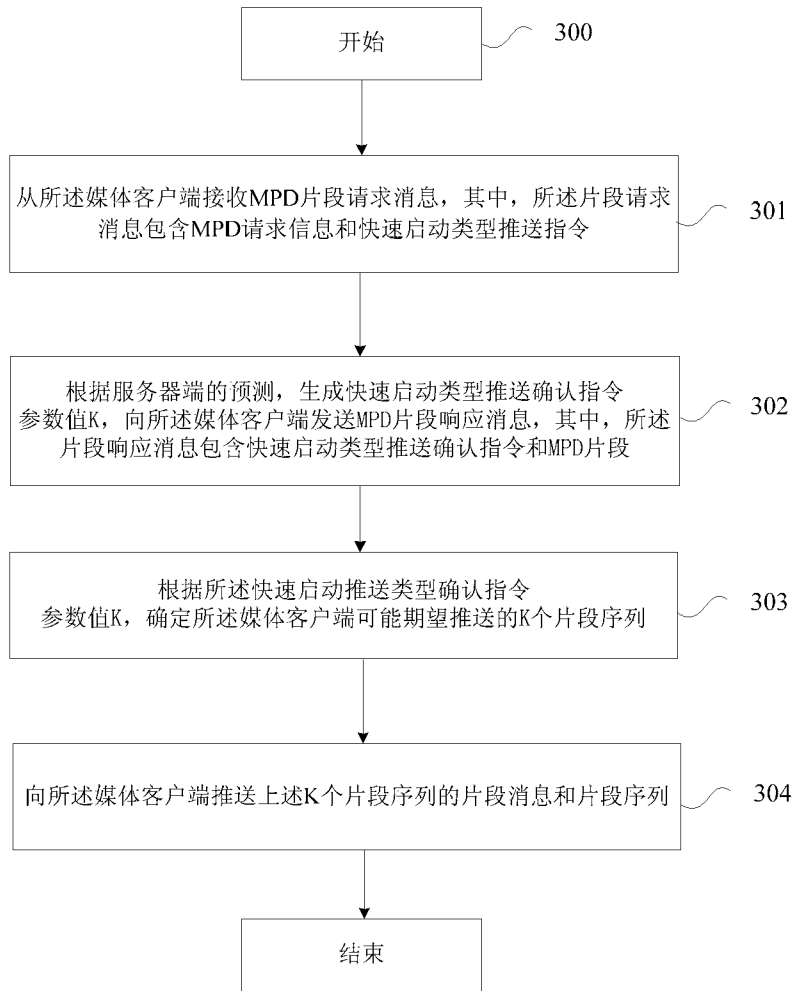


图 16



图 17



图 18

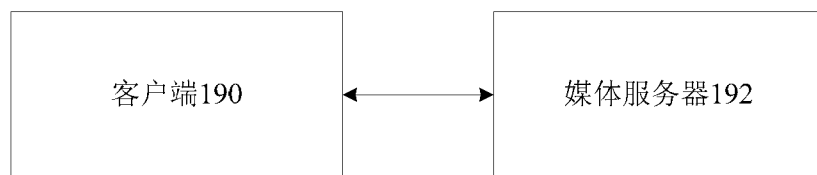


图 19

# INTERNATIONAL SEARCH REPORT

International application No.  
PCT/CN2017/086012

## A. CLASSIFICATION OF SUBJECT MATTER

H04N 21/654 (2011.01) i; H04N 21/658 (2011.01) i  
According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04N; H04L; G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNABS, CNTXT, VEN, USTXT: HTTP, DASH, MPD, URI, streaming, fast, start, push, initialization, media, segment, request, response, parameters, client, server

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 105532013 A (CANON K.K.) 27 April 2016 (27.04.2016) description, paragraphs [0532]-[0601], and figures 26-28	1-33
A	CN 1784674 A (MICROSOFT CORP.) 07 June 2006 (07.06.2006) the whole document	1-33
A	CN 104125516 A (HUAWEI TECHNOLOGIES CO., LTD.) 29 October 2014 (29.10.2014) the whole document	1-33
A	US 2014317306 A1 (FUTUREWEI TECHNOLOGIES INC.) 23 October 2014 (23.10.2014) the whole document	1-33

Further documents are listed in the continuation of Box C.       See patent family annex.

<p>* Special categories of cited documents:</p> <p>“A” document defining the general state of the art which is not considered to be of particular relevance</p> <p>“E” earlier application or patent but published on or after the international filing date</p> <p>“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>“O” document referring to an oral disclosure, use, exhibition or other means</p> <p>“P” document published prior to the international filing date but later than the priority date claimed</p>	<p>“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</p> <p>“&amp;” document member of the same patent family</p>
---	---

Date of the actual completion of the international search  
09 August 2017

Date of mailing of the international search report  
16 August 2017

Name and mailing address of the ISA  
State Intellectual Property Office of the P. R. China  
No. 6, Xitucheng Road, Jimenqiao  
Haidian District, Beijing 100088, China  
Facsimile No. (86-10) 62019451

Authorized officer  
  
ZHANG, Mingjun  
Telephone No. (86-10) 62412163

**INTERNATIONAL SEARCH REPORT**  
Information on patent family members

International application No.  
PCT/CN2017/086012

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
CN 105532013 A	27 April 2016	KR 20160032143 A	23 March 2016
		US 2016198012 A1	07 July 2016
		WO 2015004276 A3	19 March 2015
		WO 2015004276 A2	15 January 2015
		JP 2016531466 A	06 October 2016
		EP 3020208 A2	18 May 2016
CN 1784674 A	07 June 2006	ZA 200504759 A	30 August 2006
		AU 2004288602 A1	17 November 2005
		CN 1784674 B	26 January 2011
		CA 2507804 A1	03 November 2005
		CN 102123298 B	26 December 2012
		CN 102123298 A	13 July 2011
		KR 20070007698 A	16 January 2007
		RU 2005120654 A	27 April 2006
		BR PI0406598 A	09 May 2006
		US 2005262251 A1	24 November 2005
		WO 2005111837 A1	24 November 2005
		RU 2364924 C2	20 August 2009
		EP 1745387 A1	24 January 2007
		US 7720983 B2	18 May 2010
		MX PA05007086 A	23 January 2006
		JP 2007536796 A	13 December 2007
ZA 200504759 B	30 August 2006		
TW 200537293 A	16 November 2005		
EP 1745387 A4	06 April 2011		
CN 104125516 A	29 October 2014	EP 2938091 A1	28 October 2015
		US 9628547 B2	18 April 2017
		JP 2016519895 A	07 July 2016

**INTERNATIONAL SEARCH REPORT**  
Information on patent family members

International application No.  
PCT/CN2017/086012

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
		EP 2938091 A4	24 February 2016
		US 2014325018 A1	30 October 2014
		WO 2014173079 A1	30 October 2014
		KR 101734168 B1	11 May 2017
		KR 20150104188 A	14 September 2015
US 2014317306 A1	23 October 2014	US 9438654 B2	06 September 2016

国际检索报告

国际申请号

PCT/CN2017/086012

<p><b>A. 主题的分类</b></p> <p>H04N 21/654(2011.01)i; H04N 21/658(2011.01)i</p> <p>按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类</p>																	
<p><b>B. 检索领域</b></p> <p>检索的最低限度文献(标明分类系统和分类号)</p> <p>H04N; H04L; G06F</p> <p>包含在检索领域中的除最低限度文献以外的检索文献</p> <p>在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用))</p> <p>CNABS, CNTXT, VEN, USTXT; 流媒体, 快速, 启动, 推送, 初始化, 媒体, 片段, 请求, 响应, 参数, 客户端, 服务器, HTTP, DASH, MPD, URI, streaming, fast, start, push, initialization, media, segment, request, response, parameters, client, server</p>																	
<p><b>C. 相关文件</b></p> <table border="1"> <thead> <tr> <th>类型*</th> <th>引用文件, 必要时, 指明相关段落</th> <th>相关的权利要求</th> </tr> </thead> <tbody> <tr> <td>X</td> <td>CN 105532013 A (佳能株式会社) 2016年 4月 27日 (2016 - 04 - 27) 说明书第[0532]-[0601]段, 附图26-28</td> <td>1-33</td> </tr> <tr> <td>A</td> <td>CN 1784674 A (微软公司) 2006年 6月 7日 (2006 - 06 - 07) 全文</td> <td>1-33</td> </tr> <tr> <td>A</td> <td>CN 104125516 A (华为技术有限公司) 2014年 10月 29日 (2014 - 10 - 29) 全文</td> <td>1-33</td> </tr> <tr> <td>A</td> <td>US 2014317306 A1 (FUTUREWEI TECHNOLOGIES INC) 2014年 10月 23日 (2014 - 10 - 23) 全文</td> <td>1-33</td> </tr> </tbody> </table> <p><input type="checkbox"/> 其余文件在C栏的续页中列出。 <input checked="" type="checkbox"/> 见同族专利附件。</p> <p>* 引用文件的具体类型:          “A” 认为不特别相关的表示了现有技术一般状态的文件          “E” 在国际申请日的当天或之后公布的在先申请或专利          “L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件(如具体说明的)          “O” 涉及口头公开、使用、展览或其他方式公开的文件          “P” 公布日先于国际申请日但迟于所要求的优先权日的文件          “T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件          “X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性          “Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性          “&amp;” 同族专利的文件</p>			类型*	引用文件, 必要时, 指明相关段落	相关的权利要求	X	CN 105532013 A (佳能株式会社) 2016年 4月 27日 (2016 - 04 - 27) 说明书第[0532]-[0601]段, 附图26-28	1-33	A	CN 1784674 A (微软公司) 2006年 6月 7日 (2006 - 06 - 07) 全文	1-33	A	CN 104125516 A (华为技术有限公司) 2014年 10月 29日 (2014 - 10 - 29) 全文	1-33	A	US 2014317306 A1 (FUTUREWEI TECHNOLOGIES INC) 2014年 10月 23日 (2014 - 10 - 23) 全文	1-33
类型*	引用文件, 必要时, 指明相关段落	相关的权利要求															
X	CN 105532013 A (佳能株式会社) 2016年 4月 27日 (2016 - 04 - 27) 说明书第[0532]-[0601]段, 附图26-28	1-33															
A	CN 1784674 A (微软公司) 2006年 6月 7日 (2006 - 06 - 07) 全文	1-33															
A	CN 104125516 A (华为技术有限公司) 2014年 10月 29日 (2014 - 10 - 29) 全文	1-33															
A	US 2014317306 A1 (FUTUREWEI TECHNOLOGIES INC) 2014年 10月 23日 (2014 - 10 - 23) 全文	1-33															
国际检索实际完成的日期	2017年 8月 9日	国际检索报告邮寄日期	2017年 8月 16日														
ISA/CN的名称和邮寄地址	中华人民共和国国家知识产权局(ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088 传真号 (86-10)62019451	受权官员	张明俊 电话号码 (86-10)62412163														

国际检索报告  
关于同族专利的信息

国际申请号

PCT/CN2017/086012

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	105532013	A	2016年 4月 27日	KR	20160032143	A	2016年 3月 23日
				US	2016198012	A1	2016年 7月 7日
				WO	2015004276	A3	2015年 3月 19日
				WO	2015004276	A2	2015年 1月 15日
				JP	2016531466	A	2016年 10月 6日
				EP	3020208	A2	2016年 5月 18日
				-----	-----	-----	-----
CN	1784674	A	2006年 6月 7日	ZA	200504759	A	2006年 8月 30日
				AU	2004288602	A1	2005年 11月 17日
				CN	1784674	B	2011年 1月 26日
				CA	2507804	A1	2005年 11月 3日
				CN	102123298	B	2012年 12月 26日
				CN	102123298	A	2011年 7月 13日
				KR	20070007698	A	2007年 1月 16日
				RU	2005120654	A	2006年 4月 27日
				BR	PI0406598	A	2006年 5月 9日
				US	2005262251	A1	2005年 11月 24日
				WO	2005111837	A1	2005年 11月 24日
				RU	2364924	C2	2009年 8月 20日
				EP	1745387	A1	2007年 1月 24日
				US	7720983	B2	2010年 5月 18日
				MX	PA05007086	A	2006年 1月 23日
				JP	2007536796	A	2007年 12月 13日
				ZA	200504759	B	2006年 8月 30日
TW	200537293	A	2005年 11月 16日				
EP	1745387	A4	2011年 4月 6日				
-----	-----	-----	-----	-----	-----	-----	
CN	104125516	A	2014年 10月 29日	EP	2938091	A1	2015年 10月 28日
				US	9628547	B2	2017年 4月 18日
				JP	2016519895	A	2016年 7月 7日
				EP	2938091	A4	2016年 2月 24日
				US	2014325018	A1	2014年 10月 30日
				WO	2014173079	A1	2014年 10月 30日
				KR	101734168	B1	2017年 5月 11日
				KR	20150104188	A	2015年 9月 14日
-----	-----	-----	-----	-----	-----	-----	
US	2014317306	A1	2014年 10月 23日	US	9438654	B2	2016年 9月 6日

表 PCT/ISA/210 (同族专利附件) (2009年7月)